

Datacap 9.1.3

IBM

Spis treści

Dokumentacja programu IBM Datacap 9.1.3	1
Co nowego w wersji 9.1.3 programu IBM Datacap	1
Co nowego w wersji 9.1.3	2
Ułatwienia dostępu w programie Datacap Navigator	5
Skróty klawiszowe w programie Datacap Navigator	9
Przegląd	11
Datacap software components	11
FastDoc	15
Datacap capture solutions	16
Medical Claims	16
Datacap Accounts Payable	17
Locating applications on a network	17
Scenario: An imaging solution to streamline document storage	18
Scenario: Order processing automation for a mail-order pharmacy	19
Uwagi	19
Postanowienia dotyczące ochrony prywatności	22
Znaki towarowe	22
Instalowanie	23
Planowanie systemu	23
System requirements	25
Planning your system architecture	26
Application Manager	28
Rulerunner Manager	29
Datacap Server Manager	30
Domains and Windows accounts	30
Users and groups	31
Database users	31
Authentication	32
Stations	32
Databases	32
Microsoft SQL Server prerequisites	33
Oracle database prerequisites	34
DB2 prerequisites	34
Connecting to databases	35
Installation methods	36
High availability (load balancing)	37
Przygotowywanie systemu	39
Backing up and removing Datacap version 8.0.1 or earlier	40
Installation and configuration prerequisites	41
FastDoc prerequisites	41
Microsoft Internet Information Services and Microsoft .NET Framework	42
Turning on Microsoft .NET Framework 3.5.1	42

Updating ASP.NET	43
Verifying that IIS components are installed	43
Setting up the scanner	44
Ensuring the correct Windows language is selected	44
Installing Microsoft .NET Framework 4.0	45
Ensuring that the Datacap Server Service is started	45
Ensuring that the Datacap Server Service is stopped	46
Instalowanie i konfigurowanie na jednym komputerze	46
Installing FastDoc on one machine	47
Installing Datacap on one machine	48
Adding the local system account to Administrators group	49
Installing Datacap components on one machine	49
Exporting encryption keys	50
Datacap Web Client installation and configuration	50
Creating the Datacap Web Client site	51
Internet Explorer configuration for Datacap Web Client	52
Adding the tmweb.net address as a trusted site	52
Configuring Internet Explorer on a single machine	53
Datacap Report Viewer	53
Overview of Datacap Report Viewer installation and configuration	54
Adding an application pool for Report Viewer	54
Single machine installation: Setting up the Datacap Report Viewer website	55
Adding the Datacap Report Viewer address as a trusted site	55
Rulerunner Service (single thread)	56
Configure Rulerunner on a single machine	56
Granting the Rulerunner account the Log On as Service privilege	57
Configuring Rulerunner to run TravelDocs tasks	57
Configuring TravelDocs task profiles	58
Starting the Rulerunner Service	59
Monitoring batches during Rulerunner processing	59
Instalowanie i konfigurowanie w środowisku klient/serwer	59
Datacap installation and configuration in a client/server environment	60
Datacap client/server configuration	61
Installing and configuring Datacap on a client and server	62
Installation instructions for Datacap server	63
Ensure an account exists for the Datacap server service	64
Installing Datacap server	64
Configuring Datacap on the server	64
Granting Datacap server service the Log On as a Service privilege	65
Setting up sharing permissions on the Datacap folder	65
Setting up security on the Datacap folder	66
Setting up security on the Datacap\RRS folder	66
Configuring a Datacap application for Datacap Mobile	66
Configuring property display for mobile	69
Configuring bar code based classification	70
Configuring template based zonal recognition	71
Configuring recognition of identification documents	72
Configuring check recognition	75

Configuring the geolocation	76
Exporting encryption keys	77
Client/server environment: Datacap Web Client installation and configuration	78
Client/server environment: Installing and configuring Datacap Web Client on an IIS server	78
Client/server environment: Verifying that IIS components are installed	79
Client/server environment: Ensuring an account exists for Datacap Web Client	79
Client/server environment: Installing Datacap Web Client	80
Client/server environment: Importing encryption keys to Datacap computers	80
Client/server environment: Creating the Datacap Web Client site	81
Client/server environment: Setting the Datacap Web Client Application Pool Identity	82
Client/server environment: Changing the SSL setting in the server.ini file (optional)	83
Installing and configuring Datacap Navigator	83
Datacap Navigator installation steps	84
Upgrading IBM Daeja ViewONE Virtual	85
Adding applications to Datacap Navigator	86
Configuring variables in the batch structure	87
Creating a choice list for a field in the field detail panel	88
Adding Datacap Navigator tasks to your application	88
Customizing Datacap Navigator desktops	89
Configuring Internet Explorer for TWAIN scanning in Datacap Navigator	90
Running validation rules	91
Extracting data to drop-down list	91
Role-based redaction	93
Enabling Role-based redaction	94
Role-based redaction settings	95
Creating and editing role-based redaction	96
Role-based batch filtering	97
Weight for the groups	98
Assigning weight to the groups	98
Creating custom panels in Datacap Navigator	98
Adding custom code to Datacap Navigator panels	99
Adding a Start Panel widget on a Classify task page by using the Classify.js program	100
Adding a Start Panel widget on a Multiple task page by using the Multiple program	100
Changing the layout of the widgets by changing your user settings	101
Installing the developer workstation software components	101
Creating or ensuring accounts exist for developers	102
Importing encryption keys to Datacap computers	102
Setting up the scanner	103
Configure Internet Explorer on the developer workstation	103
Copying the application to the Datacap Server	104
Sharing the Datacap and tmweb.net folders on the Datacap web server	105
Starting Datacap Studio to use the Application wizard	105
Setting or ensuring the correct Datacap.xml file is in use	106
Copying the application to Datacap Server	106
Copying the Datacap.xml file from the developer workstation to the Datacap Server	109
Setting the location of the Datacap Server and the Datacap.xml file	109
Complete the Datacap server configuration	110
Updating the datacap.xml file on the Datacap server	110

Setting up security on the Datacap\Application folder	111
Complete the Datacap Web Client server configuration	112
Setting the location of the datacap.xml file	112
Restarting Internet Information Services (IIS)	112
Configuring and testing the remote workstation	112
Packaging the Datacap Web Client Configuration Tool	113
Adding the TMWeb.net address as a trusted site	114
Configuring and testing Internet Explorer using the Web Client Configuration package	114
Configuring Internet Explorer manually	115
Testing Internet Explorer	116
Installing the Datacap client on the user workstation	116
Ensure that accounts exist for users	117
Importing encryption keys to Datacap computers	117
Setting up sharing and security permissions for users	118
Setting up the scanner	119
Setting the location of the datacap.xml file	119
Installing and configuring Datacap Report Viewer	119
Datacap Report Viewer	120
Client/server environment: Overview of Datacap Report Viewer installation and configuration	120
Creating or ensuring an account exists for Report Viewer	120
Installing and configuring Datacap Report Viewer on a web server	121
Verifying that IIS components are installed	121
Setting up sharing permissions for Datacap Report Viewer on the Datacap folder	122
Setting up security for Datacap Report Viewer on the Datacap folder	122
Installing Datacap Report Viewer on the web server	123
Enabling ADSI or LDAP authentication with Report Viewer	123
Importing encryption keys to Datacap computers	124
Adding an application pool for Report Viewer	125
Client/server environment: Setting up the Datacap Report Viewer website	125
Setting the location of the datacap.xml file	126
Adding the Datacap Report Viewer address as a trusted site	126
Installing and configuring the Rulerunner Service	127
Rulerunner configuration assumptions	127
Overview of Rulerunner installation in a client/server environment	128
Configuring Rulerunner authentication	129
Creating or ensuring an account exists for Rulerunner	129
Authenticate Rulerunner Service	130
Obtaining the name of the domain	131
Obtaining the name of the AD/LDAP security group	131
Logging in to Datacap Web Client	131
Adding a Datacap group to your application for Rulerunner	132
Adding a Datacap station to your application for Rulerunner	132
Adding a Datacap user to your application for Rulerunner	133
Authenticating Rulerunner using Datacap authentication	133
Installing and configuring the Rulerunner Service	134
Shutting down the Datacap software	134
Configuring Rulerunner security and permissions on the Datacap server	135
Setting up sharing permissions for Rulerunner on the Datacap folder	135

Setting up security for Rulerunner on the Datacap folder	136
Setting up security on the RRS folder	136
Setting up security on the Datacap\Application folder for Rulerunner	137
Installing Rulerunner on the Rulerunner server	137
Installing third party software components on Rulerunner server	138
Importing encryption keys to Datacap computers	138
Setting the location of the datacap.xml file	139
Granting permissions to the Rulerunner account on the Rulerunner server	139
Setting up security on the systemprofile\AppData folder for Rulerunner	140
Granting Rulerunner the Log On as Service privilege	140
Configuring Rulerunner to run your applications	141
Gathering information that you need to set up Rulerunner	142
Configuring the task profiles for Rulerunner to run	143
Configuring Rulerunner to run tasks	144
Starting the Rulerunner Service	146
Monitoring batches during Rulerunner processing	146
Restart Datacap software	146
Rulerunner thread configuration	146
Installing and configuring the Datacap Fingerprint Service	147
Fingerprint Service configuration assumptions	149
Creating or ensuring a Fingerprint Service account exists	149
Setting up sharing permissions for the Fingerprint Service on the Datacap folder	150
Setting up security on the Datacap\application\fingerprint folder for the Fingerprint Service	150
Setting up security on the Datacap\application\batches folder for the Fingerprint Service	150
Installing the Fingerprint Service on the Fingerprint Service server	151
Setting up security on the Datacap\FingerprintService folder for the Fingerprint Service	151
Adding the Fingerprint Service account to the IIS_IUSRS group	152
Adding an Application Pool for the Fingerprint Service	152
Setting up the Fingerprint Service on the Fingerprint Service server	152
Validating the Fingerprint Service installation	153
Verifying that the Fingerprint Service can load fingerprints from your application	153
Installing and configuring the Datacap Web Client upload service	154
Editing the configuration file	156
Installing the Datacap Web Client upload service	157
Configuring applications for the Datacap Web Client upload service	158
Uploading batches scanned by using Datacap Web Client	160
Uploading batches stored locally by connecting to the Datacap server	161
Uploading completed offline batches by using FastDoc	162
Using Datacap Web Client upload service with Datacap Desktop	163
Starting the Datacap Web Client upload service	166
Viewing the Event Log	167
Disabling the Datacap Web Client upload service	167
Starting Datacap Studio	168
Installing and configuring Datacap Maintenance Manager	168
Creating or ensuring an account exists for Datacap Maintenance Manager	169
Setting Datacap Maintenance Manager account permissions for sharing	170
Setting Datacap Maintenance Manager account security permissions on the Datacap folder	170
Setting Datacap Maintenance Manager account security permissions on the Datacap\RRS folder	171

Installing the developer workstation software components	171
Importing encryption keys to Datacap computers	172
Configuring authentication for Datacap	172
Datacap Server service settings	174
Advanced settings for the Datacap Server service	175
TMA authentication system	176
Active Directory ADSI and LDAP authentication systems	178
ADLDS and LLDAP authentication systems	182
LLLDAP group authentication	185
Configuring the Datacap Server service to use an external authentication system	187
Authenticating Datacap Web Client users with IBM Security Access Manager	189
Configuring single sign-on (SSO) for Datacap Navigator	191
Installing and configuring IBM Datacap Advanced Handwriting Recognition 9.0.1	192
Installing IBM Datacap Advanced Handwriting Recognition 9.0.1	192
Using IBM Datacap Advanced Handwriting Recognition 9.0.1	193
Migrating from ICR_P to Advanced Handwriting Recognition actions	193
Datacap Web Services installation steps	194
Datacap Web Services authentication	196
Configuring Datacap Web Services authentication	197
Configuring Datacap Web Services ADSI or LDAP authentication	197
Configuring Datacap Web Services ADLDS or LLDAP authentication	198
Setting up sharing permissions for Datacap Web Services	199
Setting up shared folder security for Datacap Web Services	199
Setting up application folders security for Datacap Web Services	200
Installing Datacap Web Services	200
Datacap Web Services hosting options	201
Ensuring the required IIS components are installed	201
Setting up the Windows service to host Datacap Web Services	202
Setting up IIS to host Datacap Web Services	203
Disabling security for the transaction endpoints	204
Enabling SSL for Datacap Web Services	204
Importing encryption keys to Datacap computers	206
Client/server installation checklist	207
Datacap server setup	208
Install the Datacap server	208
Configuring Datacap on the server	209
Configuring Datacap Web Server on a supported version of Windows Server	211
Datacap developer workstation setup	213
Add tmweb.net address as trusted site	214
Configuring and testing IE manually	214
Copy the application to the Datacap server	215
Completing Datacap server setup	217
Update Datacap.xml on the server	217
Set application folder security permissions	217
Complete the Datacap Web Client setup	218
Running a Datacap client on Developer Workstation	218
Remote workstation setup	219
Packaging the Web Client Configuration tool	219

Add tmweb.net address as trusted site	220
Configuring and testing IE using a package	220
Configuring and testing IE manually	221
Run the Datacap Web Client application	222
User workstation and permissions setup	223
Datacap Report Viewer setup	224
Viewing Datacap Report Viewer reports	227
Rulerunner installation and configuration	228
Configure Rulerunner account permissions	230
Configure Rulerunner authentication	230
Configure Rulerunner to run application tasks	231
Fingerprint Service setup	233
Test Fingerprint Service setup	235
Datacap Maintenance Manager setup	235
Datacap installation command-line parameters	236
Viewing the installation program parameters	237
Microsoft Windows Installer parameters	237
Commonly used Datacap Setup.exe parameters	237
Konfigurowanie bazy danych Datacap	239
Configuring an application to use the database	241
Database security permissions	241
Defining the database structure	243
Creating a single DB2 database for a Datacap application	243
Creating a single Oracle database for a Datacap application	245
Creating a single Microsoft SQL Server database for a Datacap application	246
Converting applications from Jet database to Oracle database	247
Moving the application files	249
Verifying the database connection	250
Advanced database settings for Datacap Server	250
Application settings - {default} option	252
Aktualizowanie	253
Creating a test environment	253
Upgrading Datacap applications from Taskmaster 8.0.1	254
Upgrading Datacap applications from Taskmaster 8.1	257
Updating Microsoft Access database schemas from Taskmaster 8.1	259
Upgrading Datacap applications from 9.0	259
Rebuilding customized screen panels for Datacap Desktop	261
Repairing a Datacap 9.0.1 installation	262
Updating Microsoft Access database schemas from Datacap 9.0	263
Upgrading Datacap applications from Datacap 9.1.1	263
Migrating Datacap Navigator custom panels to Datacap 9.0 Feature Pack 2 or later	264
Setting Datacap Navigator default layouts in Datacap 9 and Feature Packs 1 and 2	265
Upgrading Datacap applications to 9.1	265
The Datacap Desktop panels are for use with IBM Datacap 9.1	266
Customized panel conversion to Datacap Desktop	267
Generating the layout XML file	267
The layout XML file	268
Creating the Datacap Desktop in Microsoft Visual Studio	268

Creating a production environment	269
Deinstalacja programu Datacap	269
Backing up custom applications	270
Backing up custom application shortcuts	271
Removing Datacap Web Client and Datacap Report Viewer sites	271
Removing the Datacap Application Pools	271
Removing IBM Datacap Version 9.0.1	272
Removing accounts from Administrator and Backup Operator groups	272
Deleting the Datacap registry keys	272
Deleting the program shortcuts	273
Deleting the remaining folders	273

Monitorowanie

Monitoring your system with Datacap Navigator	273
Accessing the dashboard	274
Enabling application statistics collection	274
Configuring dashboard features	275
Enabling email notification	275
Dashboard parameters	276
Monitoring your system with Datacap Report Viewer	278
Logging in to Datacap Report Viewer	278
Viewing a standard Datacap Report Viewer report	279
Viewing a custom report in the Datacap Report Viewer web interface	279
Creating a Datacap Report Viewer report filter	280
Adding reports to the Datacap Report Viewer dashboard	280
Modifying the Datacap Report Viewer dashboard	281
Standard reports	281

Dostosowywanie i uruchamianie aplikacji

Running tasks with Datacap Desktop	283
Task profiles	284
Starting a task	284
Running the VScan task	285
Running the Scan task	285
Running the Fixup task	286
Preparing and running FastDoc applications	287
Getting started on FastDoc	287
Starting FastDoc	288
Creating an application in Application Wizard	289
Forms template application configuration	290
Jobs available in Forms template applications	290
Setting up documents on Forms template applications	291
Setting up image enhancement on Forms template applications	292
Setting up fingerprints on Forms template applications	292
Setting up field recognition on Forms template applications	293
Setting up field validation on Forms template applications	294
Learning template application configuration	294

Jobs available in Learning template applications	295
Setting up fields on Learning template applications	296
Setting up image enhancement on Learning template applications	296
Locating fields in Learning template applications	297
Setting up field validation in Learning template applications	297
Configuring export options	297
Format of the exported data file	298
Confirming a successful export	299
Application configuration on FastDoc	299
Preparing for document scanning and indexing	300
Setting up a scanner for FastDoc	301
Setting up batch profiles in Local mode	301
Setting up batch profiles in Datacap Server mode	303
Adding rulesets to a FastDoc workflow	303
Configuring rulesets for the application	304
Testing rulesets	306
Defining an index field with keywords	306
Defining index field validation by using a database	307
Using Click N Key to capture data	307
Document processing on FastDoc	308
Preparing paper documents for scanning	308
Running FastDoc in Local mode	309
Opening an existing batch	310
Creating a batch with pre-scanned images	310
Creating a batch by using a scanner	311
Indexing and verifying batches	312
Running FastDoc in Datacap Server mode	313
Processing documents in Forms template applications	314
Processing documents in Learning template applications	315
Displaying a page or document	316
Deleting pages	317
Accounts Payable application	317
Setting up Datacap Accounts Payable to process your invoices	317
Adding your vendors to the Demo vendors list	318
Adding your purchase order line items to a simulated database	319
Datacap Accounts Payable client procedures	320
Scanning invoice images on Datacap Desktop	321
Preparing invoice images for processing on Datacap Desktop	322
Processing invoice images on Datacap Desktop	323
Verifying invoice data in Datacap Desktop	323
APT Verify window instructions	326
Handling an unknown invoice when the vendor is unknown	327
Identifying detail lines on an unknown invoice	328
Processing multiple unknown invoices for a known vendor in a single batch	328
Associating a vendor with a fingerprint	328
Capturing fields on a rotated image	329
Permission denied: LoadPicture Error	329
Changes to Pre-Existing Zone Positions Are Ignored	330

Identify Fingerprint ID Associated with Problem Invoice	330
Exporting invoice images on Datacap Desktop	331
Datacap Accounts Payable Web Client procedures	331
Review default installation results	332
Starting Datacap Accounts Payable on Datacap Web Client	333
Viewing the default user security group on Datacap Web Client	333
Viewing the default user permissions on Datacap Web Client	334
Viewing the default station permissions on Datacap Web Client	334
Logging in to Datacap Accounts Payable on Datacap Web Client	335
Scanning invoice images on Datacap Web Client	335
Uploading invoice images on Datacap Web Client	336
Processing invoice images on Datacap Desktop	337
Verifying invoice data on Datacap Web Client	337
Exporting invoice images on Datacap Desktop	339
Medical Claims application	340
Medical Claims application client procedures	340
Virtually scanning claim form images using Datacap Desktop	342
Scanning physical claim forms using Datacap Desktop	342
Running the Medical Claims Background task using Datacap Desktop	343
Fixing Medical Claims document integrity problems using Datacap Desktop	343
Verifying scanned claims using Datacap Desktop	344
Processing your own claim form images	346
Medical Claims application in Datacap Web Client	346
Logging in to Medical Claims in Datacap Web Client	347
Viewing Medical Claims permissions in Datacap Web Client	348
Scanning claim forms in Datacap Web Client	348
Uploading Medical Claims scanned images in Datacap Web Client	350
Verifying Medical Claims data in Datacap Web Client	350
TravelDocs application	352
Accessing the Datacap Web Client Login page and logging in	353
Running the TravelDocs VScan task	353
Running the TravelDocs Web VScan task	354
Running the TravelDocs Web Upload task	355
Running the TravelDocs PageID task	355
Running the TravelDocs Batch Profiler task	356
Running the TravelDocs Verify task	356
Running the TravelDocs Web Verify task	357
Running the TravelDocs Export task	357
Konfigurowanie składnika Rulerunner do wykonywania czynności w aplikacji	358
Konfigurowanie czynności skanowania tak, aby puste strony były pomijane	358

Administrowanie

Datacap web clients administration	359
Users, groups, and stations administration	361
Jobs and tasks administration	361
The Job Monitor and workflow administration	362
Shortcuts administration	362
Batch queuing to specific users and stations	363

Effects of application pool recycling on Datacap Web Client batches	364
Creating a custom column in the Job Monitor	365
Administering Datacap Navigator	365
Datacap Navigator user settings	366
Administration view	366
Batch status (Job Monitor)	367
Batch structure labels	367
Speed Scan and Speed Index	368
Setting up Datacap Navigator and custom panels	371
Batch Preparation client	372
Creating a batch by using Batch Preparation client	372
Speed Scan client	372
Thumbnail view	373
Thumbnails - moving and deleting	373
Undocking Viewer to second monitor	374
Persistence of the panel arrangement or sizing	374
Tab order customization	375
Scanner properties	375
Security-driven access	377
Creating custom hot keys for toolbar buttons	377
Using asynchronous upload feature	378
Speed Index Client	378
Index fields display customization	379
Graphical forms designer	379
Customizing document fields using Document Panel	380
Adding button to toolbar menu	381
Using lookup button for populating document fields	381
Deploying SOAP web service lookup	382
Deploying MSSQL lookup	383
Deploying lookup button in custom panel: Example 1	384
Deploying lookup button in custom panel: Example 2	385
External Data Services	385
Deploying External Data Services on WebSphere Application Server console	386
Support for ADLDS	387
Support for skipping blank pages	387
Image scans	388
Batch verification	388
Batch upload	389
Page classification	389
Batch processing (Task List)	390
Setting Datacap Navigator default page layouts	390
Constructing a URL for Datacap Navigator	391
External data services for Datacap Navigator	395
Customizing Job Monitor	397
Datacap Navigator access	398
Datacap Desktop administration	398
Creating and configuring a task to use with Datacap Desktop	399
Using command line parameters for Datacap Desktop	400

Setting up a barcode type as a document separator	401
Configuring the image selection mode for the Datacap Desktop Fixup task	402
Datacap Application Copy Tool	402
Configure the connection strings	403
Datacap Application Copy Tool Command Line Interface	404
Application migration options	406
Moving an application into a new environment	408
Updating an application in an existing environment	409
Changing the database provider	410
Migrating a single database to another database provider	411
Datacap Application Copy Tool User Interface	412
Monitoring system performance with IBM System Dashboard for Enterprise Content Management	413
Shutting down Datacap for maintenance	417
FastDoc maintenance	418
Purging finished batches	418
Deleting selected batches	419
Viewing daily batch statistics	419
Determine the original file name of a pre-scanned image	420
Maintaining fingerprints by using the Fingerprint Maintenance Tool	420
Starting the Fingerprint Maintenance Tool	421
Locating and deleting partial fingerprints	421
Deleting fingerprints	421
Deleting fingerprints that are associated with a document type	422
Exporting selected fingerprints	422
Adding selected fingerprints	422
Troubleshooting the Fingerprint Maintenance Tool	423
FMT Backup Directory	423
FMT.Log	423

Tworzenie aplikacji	423
Getting started with application development	424
Datacap application development	425
Business Requirements and Application Architecture	427
Business requirements development	427
General Datacap application architecture	428
TravelDocs: Business requirements	429
Document types and page types	429
Required document structure	433
Fields for each page type	434
Permissible field values	435
Business validation rules	435
Data export format	436
Datacap Studio	437
Quick tour of the user interface	438
Starting Datacap Server	438
Opening a sample Datacap application	439
Panel organization within Datacap Studio	439
The Rulemanager tab	440

The Zones tab	440
The Test tab	441
TravelDocs: Start the TravelDocs application	441
The application framework	442
Connecting to the application	442
Document hierarchy	443
Document structure	443
Identification of page types from documents	444
Relation of the document hierarchy to the runtime batch hierarchy	444
Page type versions	445
TravelDocs: Create the document hierarchy	445
Default document hierarchy	446
Creating document types	446
Creating page types	446
Specifying the structure of documents and pages within the batch	447
Creating data fields	449
Specifying the structure of fields on each page	450
Sharing field definitions across the document hierarchy	450
The Datacap workflow	451
Understanding the Datacap workflow	451
Workflows, jobs, and tasks	452
Task profiles and rulesets	453
Rulesets, rules, and actions	453
Document input	454
Electronic document input (virtual scanning)	454
Document conversion	455
Hardcopy document scans	455
Local scanning	456
Remote scanning	456
TravelDocs: Batch creation with VScan	456
Scanning the sample documents from the application images folder	457
Modifying the VScan ruleset	457
Running VScan to generate a batch	457
Examining the files in the runtime batch folder	458
Local scanner setup (optional)	459
Creating the scan task in the Datacap Web Client	460
Creating a shortcut for the new scan task	460
Running the scan task	461
Page Identification	461
Page identification methods	462
Fingerprint matching	462
Structure-based page identification	464
Text matching	465
IBM Content Classification: Category and rule-based classification	465
Manual page identification	466
Image Enhancement	466
Goal of image enhancement	466
When to complete image enhancement	467

TravelDocs: Fingerprint library creation	467
Changing the fingerprint creation method	467
Fingerprint creation for known page types	468
Creating fingerprint classes	468
Adding individual fingerprints	469
TravelDocs: Sample fingerprint image enhancement	469
Determining appropriate image-processing settings	470
Applying new image-processing settings to enhance the fingerprint images	471
TravelDocs: Run a batch through the workflow	471
Processing a batch	472
Runtime batch folder contents	472
Checking the confidence levels on the runtime pages	473
Rule Execution	473
Association of rules with objects	474
Example 1: Batch-level rule execution	475
Example 2: Page-level rule execution	475
Order of rule execution	475
Example 1: Page identification rules	477
Example 2: Validation rules	478
Summary of order of rule execution	478
Running rules directly on images by using Datacap Web Services transactional endpoints	479
Setting registry keys for Transaction/Execute	480
TravelDocs: Stepping a batch through the PageID task profile	481
Document assembly	481
Structured documents	482
Hierarchy-based documents	482
Assembling documents	483
Creation of the page data files	484
Document integrity	485
CheckAllIntegrity action	486
Document integrity problem management	486
TravelDocs: Document creation and page file setup	487
Running a batch through the workflow	487
Contents of the runtime batch folder	488
Page data files	488
TravelDocs: Document integrity management	489
Configuring branching	489
Running a batch with document integrity problems	490
Data recognition	490
Page data recognition	491
Identifying recognition zones by using fingerprints	492
Recognition zone information storage	492
Reading data from the page	492
Dynamic locale support	493
Setting locale values	494
Recognition language settings	495
Enabling automatic language detection for OCR_A recognition	496
Supported language codes	498

Check box options management	501
Check box recognition methods	501
Establishing parent fields	502
Setting the required variables on the parent field	503
Implementing the OCR/A check box recognition method	503
Using the pixel threshold evaluation method	503
Identifying medical claim forms by using Autofield	505
TravelDocs: Specification of recognition zones	506
Creating the text zones on the Rental_Agreement page	506
Creating the OMR zones on the Rental_Agreement page	506
Creating the zones for the other page types	507
TravelDocs: Assignment of default rules to the document hierarchy	508
Assigning the default page level rules to new pages	508
Assigning the default field level rules to new fields	509
Updating the Recognize Page rule	509
Running a batch through the workflow	510
TravelDocs: Updating the application to manage check box options	510
Setting the required variables on the Options and Insurance fields	511
Specifying the check mark type	511
Creating a rule to recognize the OMR fields	512
Adding the Recognize OMR Fields rule to the document hierarchy	512
Running a batch through the workflow	512
TravelDocs: Using pixel threshold check box recognition (optional)	513
Updating the Recognize OMR Fields rule to use RecogOMRThreshold	513
Determining appropriate threshold and background settings	514
Checking the option values and obtaining the density string values	514
Interpreting the density string values	515
Data Validation	516
Validate the data	516
Check data format validity	517
Validate calculated fields	518
Show validation failures to an operator	519
Use external data sources during validation	520
Manage validation errors	521
TravelDocs: Update the application to complete validation	521
Validate the currency fields	522
Creating the Validate Currency Field rule	522
Adding the Validate Currency Field rule to the document hierarchy	523
Validate the flight cost	523
Creating the Validate Flight Cost rule	524
Adding the Flight Cost rule to the document hierarchy	524
Use a lookup database to validate the car type	525
Creating the lookup database table	525
Creating the Validate Car Type rule	526
Adding the Validate Car Type rule to the document hierarchy	526
Creating a dictionary of valid car types	526
Creating the dictionary	527
Attaching the dictionary to the Car_Type field	527

Running a batch through the workflow	528
Examination of page and field status values	528
Creating recognition zones for the remaining fingerprints	531
Running a batch through the workflow	532
Page and field status codes in the TravelDocs application	532
Data verification	533
Field data verification	533
Options for data verification	534
Confidence levels and the page status	534
Confidence levels	535
Page status	536
Overriding the default confidence value on specific fields	536
Overriding validation failures	537
Skipping a verification task	537
TravelDocs: Batch verification	538
Setting the Car Type field to prevent overriding	539
Batch verification with Datacap Desktop	539
Creating dictionaries for check box options	539
Preparing a batch for verification	540
Opening the batch in Datacap Desktop	540
Reviewing the batch in Datacap Desktop	541
Submitting the batch	541
Verifying batches with Datacap Web Client	541
Data export	543
Exporting data	544
Export to a text file	544
Configure text export for IBM Content Manager OnDemand	545
Export to a database	545
Export to an XML file	546
Datacap Connector actions	547
Verifying the installation	548
Content repository authentication	548
Integrating Connector actions into applications	549
Storing passwords in the .app file	550
Connector actions configuration	550
IBM Content Manager Connector actions	551
IBM Content Manager Connector prerequisites	551
IBM Content Manager Connector settings	552
Configuring IBM Content Manager Connector actions	553
IBM Content Manager Connector upload examples	554
Search and download action attributes of IBM Content Manager	555
Updating IBM Content Manager content with search and download actions	556
FileNet P8 Connector actions	556
FileNet P8 Connector prerequisites	557
FileNet P8 Connector settings	558
Configuring FileNet P8 Connector actions	559
FileNet P8 Connector upload examples	560
Downloading bulk FileNet P8 content to Datacap by using FileNet Sweep Job	562

Creating JavaScript sweep action handler to download document content	563
Creating custom sweep action using the JavaScript	565
Creating custom job sweep referring the action	565
Designing your Datacap application	566
Documentum Connector actions	567
Documentum Connector prerequisites	567
Documentum Connector settings	568
Configuring Documentum Connector actions	568
Documentum Connector upload examples	569
SharePoint Connector actions	571
SharePoint Connector prerequisites	571
SharePoint Connector settings	572
SharePoint and Datacap	572
Configuring SharePoint Connector actions	574
SharePoint Connector upload examples	575
FileNet Image Services Connector Connecting actions	576
FileNet Image Services Connector prerequisites	577
FileNet Image Services Connector settings	577
Configuring FileNet Image Services Connector actions	578
FileNet Image Services Connector upload examples	579
Email Connector actions	580
Email Input actions	581
Email Send actions	582
Email Connector prerequisites	582
Email Connector settings	583
Configuring Email Connector actions	585
Email Connector import examples	586
Fax Connector actions	587
Fax Connector prerequisites	587
Fax Connector settings	588
Configuring Fax Connector actions	588
Fax Connector import examples	589
Box Connector actions	590
Configuring Box Connector actions	590
Box Connector settings	591
Box Connector upload examples	593
Connector actions log files	593
Viewing action details	594
TravelDocs: Exporting data to a database	595
Configuring the export database	595
Creating the ExportDB ruleset	595
Adding theExportDB ruleset to the Export task profile	596
Attaching the Export Rental Agreement Data rule to the rental agreement page	596
Running a batch through the workflow	597
TravelDocs: Exporting data to an XML file	597
Creating the ExportXML ruleset	598
Adding theExport XML ruleset to the Export task profile	599
Attaching the Export XML rules to the document hierarchy	599

Running a batch through the workflow	600
Application Debugging	601
Datacap log files	601
Enable logging for Datacap Web Client tasks	601
Rulerunner Service (RRS) log files	602
Set Rulerunner logging by application and task	603
Task log files	604
Debug your application from the Datacap Studio Test tab	604
Using breakpoints	605
Breakpoint types	605
Setting breakpoints	605
Disable and clear breakpoints	606
Set generic breakpoints	606
Single-stepping through your code	607
Examining log files from the Test tab	607
Handling line item grids	607
Defining the document hierarchy for line item grids	608
Rules to recognize line items	609
Text matching to locate fields	610
Removing non-line items from the page data file	611
Exporting data from a line item grid	613
TravelDocs: Adding new pages that contain line item grids	613
Updating the document hierarchy	613
Adding pages to the document hierarchy	614
Creating data fields	614
Attaching the existing page rules to the new pages	616
Creating the page fingerprints	616
Defining the recognition zones	616
TravelDocs: Recognizing line item grid data	617
Creating the recognition rules for the line items	618
Creating the recognition rule for the grid total	618
Attaching the rules to the document hierarchy	619
Running a batch through the workflow	619
Creating rules to remove the non-line items	620
TravelDocs: Validating line item grid data	621
Validating the line item totals	621
Creating the validation rule	622
Attaching the validation rule to the document hierarchy	622
Validating the grid total	622
Creating the validation rule	623
Attaching the rule to the document hierarchy	623
Running a batch through the workflow	624
TravelDocs: Verifying the line item grid pages	625
Verifying pages by using Datacap Desktop	625
TravelDocs: Exporting line item grid data to a database	626
Exporting to a database	626
Creating the export database table	626
Adding rules to the ExportDB ruleset	626

Attaching the Export Other rules to the document hierarchy	628
Running a batch through the workflow	628
Smart parameters	629
General structure of a smart parameter	630
Special variables to access application configuration settings	631
Determining the correct key name	632
Storing passwords, connection strings, and other parameters in the .app file	633
Reference passwords, connection strings, and other parameters from your actions	634
Access to the runtime hierarchy	635
Examples of using special variables to access the runtime hierarchy	635
Summary of special variables for accessing the runtime hierarchy	636
Use navigation elements to access the runtime hierarchy	638
Use other special variables	638
Access job and task information	639
Access other information	639
TravelDocs: Exporting line item grid data to an XML file	639
Adding rules to the ExportXML ruleset	640
Attaching the Export Other XML rules to the document hierarchy	641
Running a batch through the workflow	642
Text matching	643
Identify pages with text matching	643
Locate data with text matching	644
Locate simple strings	644
Use regular expressions	645
Text matching with keyword lists	645
Locate the field data	646
Update the runtime data file with the recognized text	647
Text matching for data recognition limitations	648
TravelDocs: Update the application to use text matching	648
Identifying unrecognized pages by using text matching	649
Recognizing data with text matching	650
Attaching the rules to the document hierarchy	652
Running a batch through the workflow	652
Pattern Matching	653
Pattern matching overview	653
Considerations for using pattern matching	654
Auto registration with the FindFingerprint action	655
Anchor objects setup	656
Confidence level setup for pattern matching	656
Geometric pattern matching	657
How the PatternMatch_Identify action works	657
Multiple anchor objects	658
pat_RegisterZones action to adjust the positions of individual fields	659
Text-based pattern matching	660
How the pat_RecogMatch_Id action works	661
Determine the runtime field positions by using anchor offsets	662
Field adjustment that is based on multiple anchors	662
TravelDocs: Use geometric pattern matching to identify pages	663

Setting up the pattern match anchor objects	663
Updating the PageID rule to use pattern matching	663
Running a batch through the workflow	664
Reviewing the runtime batch files	665
Workflow automation, routing, and automatic fingerprint generation	666
Use Rulerunner to automate background tasks	667
Rulerunner overview	667
Rulerunner configuration	667
Rulerunner operation	668
Rulerunner logging	668
Conditional branching and splitting to route documents	668
Branching versus splitting	669
Condition flags	670
Defining a condition and the associated action	671
Jobs to handle special conditions	672
Creating a job and task	672
Automatic fingerprint generation	673
TravelDocs: Automated background processing with Rulerunner	674
Defining background tasks in Datacap Application Manager	674
Setting up background tasks in Rulerunner Manager	675
Enabling Rulerunner logging	675
Setting up the Job Monitor	676
Running a batch through the workflow	676
Analyze the Rulerunner log	677
Disabling Rulerunner logging	678
TravelDocs: Handle document integrity failures	678
Moving document creation and integrity checking into the PageID task profile	679
Creating the CreateDocs task	679
Configuring Rulerunner to run CreateDocs	680
Running a batch through the workflow	680
TravelDocs: Identify pages manually	681
Adding a function for manual page identification	682
Updating the Recognize Page ruleset	683
Adding the conditional branch to the PageID task	684
Creating the ManualPageID job and task	684
Configuring branching and creating a shortcut	685
Configuring the Routing ruleset to handle manually identified pages	686
Running a batch through the workflow	686
Recognizing the data on the unidentified page	687
TravelDocs: Generating fingerprints automatically	688
Creating the AutoFingerprint ruleset	688
Assigning the rule to each page type	689
Adding the ruleset to the Verify task profile	690
Enabling logging for Datacap Web Client	690
Running a batch through the workflow	690
Reviewing the RRS log file	691
TravelDocs: Splitting a document from the main batch	692
Updating the Routing ruleset to split the batch	692

Assigning the Batch Splitting rule to the Close element of the batch	693
Routing the split document to a supervisor	694
Creating the supervisor job	694
Configuring the job router	694
Configuring the supervisor shortcuts	695
Running a batch through the workflow	696
Datacap Web Client and remote scanning	696
Moving the workflow to Datacap Web Client	697
Scanning images remotely	698
Configuring the remote scanning client	699
Implementing a start panel	699
Populating drop-down lists on a start panel	700
Running validation rules	701
Remote virtual scanning	701
Verification by using the VeriFine web client	702
Restructure the batch by using the batch tree view (VeriFine)	702
Configuring the VeriFine client	703
Configuring additional VeriFine settings	704
Creating custom pages	705
Verification, page identification, and registration by using AIndex	706
Restructure the batch by using the batch tree view (AIndex)	706
AIndex client configuration	707
Verifying in multiple passes	708
Storing multiple values in the runtime page data file	708
Actions that support multi-pass verification	709
Settings that support multi-pass verification	710
Example of two-pass data entry	710
Example of double-blind data entry	711
Manual page identification and registration	712
Enabling manual page registration (manual anchoring)	713
Registering a page by using manual anchoring	713
Verification by using the AVerify web client	714
Creating and using custom (static) panels	715
Exporting the default panel layout	716
Customizing the panel layout	717
Specifying the custom panels to use in a task	717
Verification by using the ImgEnter web client	718
Manual page identification and batch restructuring with ProtoId	718
ProtoID web client configuration	719
Administering an application	721
Job monitoring	721
TravelDocs: Scanning from Datacap Web Client	722
Creating a remote scan task	722
Configuring the remote scanning client	723
Configuring the Upload task	723
Scanning and uploading a batch	724
Creating the web Job CreateDocs task	725
Configuring Rulerunner to run web jobs	725

Modifying the Verify shortcut	726
Opening the batch for verification	726
TravelDocs: Using AIndex for manual page identification and registration	727
Making a copy of the application	727
Updating the application	727
Updating ManualPageID	729
Ignored field statuses	729
Done field statuses	730
Done page statuses	730
Validation statuses	730
Editing the ManualPageID settings	731
Creating the ManualIDValidate rule	731
Running a batch through the workflow	732
Testing the ManualIDValidate rule	733
Filter batches by group in the Job Monitor (Datacap Web Client)	733
Defining group names for filtering batches	734
Assigning a group to a batch for filtering	735
Configuring the filter method	735
Fingerprint Management	736
Review of basic fingerprint functionality	736
Create fingerprint files	737
Add fingerprints to the fingerprint library	737
Define field zones	737
The Fingerprint database	738
Using fingerprint XML files	739
The fingerprint XML file	739
Enable FPXML	740
Adding fingerprints using the Datacap Studio Zones tab	740
Add fingerprints using actions	741
Exporting existing position information from the document hierarchy	741
Setting up the Fingerprint Maintenance Tool for your application	742
Exporting the position information	742
TravelDocs: Updating auto fingerprinting to use FPXML	742
Updating the AutoFingerprint ruleset	743
Updating the Recognize Page rule	743
Preparations for running a batch through the workflow	744
Running a batch through the workflow	744
Configuring content classification for XML layout block parsing	745
Application translation	747
Creating a Datacap Maintenance Manager application	748
Starting Datacap Studio to use the Application wizard	749
Creating a Datacap Maintenance Manager application on Datacap Studio	750
Updating the datacap.xml file on the Datacap server	750
Setting the location of the Datacap Server and the Datacap.xml file	751
Setting Datacap Maintenance Manager account security permissions for the Datacap\NENU application folder	752
Setting Datacap Maintenance Manager account security permissions for the monitored application folder	752
Opening the Datacap Maintenance Manager application	753

Deleting rulesets from a Datacap Maintenance Manager application	753
Adding actions to a Datacap Maintenance Manager application	754
Running a rule set with Datacap Maintenance Manager	755
Configuring Windows Task Scheduler to automatically run a ruleset	756
AutoDelete batches with Datacap Maintenance Manager	757
AutoDelete Process	757
Sample AutoDelete ruleset	758
Datacap object API reference	759
Relationship between runtime batch and document hierarchy	760
Creating a document hierarchy with Datacap object APIs	761
DCO APIs	763
DCO properties	765
AltConfidenceString property	766
AltText property	766
CharConfidence property	767
CharValue property	768
ConfidenceString property	769
ID property	770
ImageName property	771
Status property	771
Text property	772
Type property	773
Variable property	773
XML property	774
DCO methods	775
AddChild method	778
AddValue method	779
AddVariable method	780
AddVariableFloat method	782
AddVariableInt method	782
AddVariableString method	783
CheckIntegrity method	784
Clear method	785
CreateDocuments method	786
CreateFields method	786
DeleteChild method	787
DeleteValue method	788
DeleteVariable method	789
FindChild method	790
FindChildIndex method	791
FindRouteChild method	792
FindVariable method	792
get_AltConfidenceString method	793
get_AltText method	794
get_CharConfidence method	795
get_CharValue method	796
get_OMRValue method	797
get_Variable method	798

GetChild method	799
GetLastError method	800
GetPosition method	801
GetRoute method	802
GetVariableName method	803
GetVariableValue method	804
IsError method	804
IsRoute method	805
IsValid method	806
MoveChild method	807
MoveIn method	808
NumOfChildren method	808
NumOfVars method	809
ObjectType method	810
Parent	811
Read method	811
ReadSetup method	813
SetPosition method	814
SetupNode method	815
SetupObject method	815
set_AltConfidenceString	816
set_AltText method	817
set_CharConfidence method	818
set_CharValue method	819
set_OMRValue	820
set_Variable method	821
Write method	822
WriteSetup method	823
DCOSetup API	824
DCOSetup properties	826
DictionaryName property	826
Path property	827
Value property	828
Word property	828
DCOSetup methods	829
AddNode method	830
DeleteNode method	831
DeleteNodeByName Method	832
get_DictionaryName method	833
get_Value method	834
get_Word method	835
GetNode method	835
GetNodeByName method	836
NumOfDictionaries method	837
NumOfNodes method	837
NumOfWords	838
ReadLock	839
ReadSetup	840

set_DictionaryName method	840
set_Value method	841
set_Word method	841
ShowSetupDialog	842
UnlockIt	842
WriteSetup method	843
DCOSetupNode APIs	843
DCOSetupNode properties	844
Name property	845
ObjectType property	846
RuleChildName	847
RuleMaxNum	847
RuleMinNum	848
RuleObjectType	849
RulePosition	850
Variable	850
VariableName	851
VariableValue	852
DCOSetupNode methods	852
AddRule method	854
AddVariable method	856
DeleteRule method	857
DeleteVariable method	857
DeleteVariableByName method	858
FindRule method	858
get_RuleChildName method	859
get_RuleMaxNumber Method	860
get_RuleMinNumber method	860
get_RuleObjectType method	861
get_RulePosition method	862
get_Variable method	863
get_VariableName method	864
get_VariableValue Method	865
GetRule method	865
NumOfRules method	866
NumOfVariables Method	867
set_RuleChildName	868
set_RuleMaxNumber	868
set_RuleMinNumber	869
set_RuleObjectType	869
set_RulePosition	870
set_Variable	870
set_VariableName	871
set_VariableValue	872
Integracja z innymi produktami	872
IBM Content Navigator and Case Manager transactional capture	872
Sample workflows	873

Configuring your workstation for scanning	873
Scanning documents in IBM Content Navigator	874
Scanning documents in IBM Case Manager	874
Adding documents to a batch from an IBM Content Navigator repository	875
Configuring Datacap for transactional capture	875
Configuring scanning in IBM Content Navigator	876
Configuring scanning in IBM Case Manager	877
Configuring IBM Content Navigator for adding documents to batches	878

Rozwiązywanie problemów i wsparcie 878

Troubleshooting process overview	879
Searching knowledge bases	881
Getting fixes from Fix Central	882
Contacting IBM Software Support	882
Subscribing to Support updates	884
Troubleshooting tips for IBM Datacap	885
Invalid character when scanning document using Dynamic Web TWAIN HTML5 driver	885
Scanner settings do not persist when scanning using Dynamic Web TWAIN Interface	887
After image rotation, snippet shows image in its original orientation	888
AppScan issues: Physical path disclosure and hidden directory detection	888
Document integrity check fails to identify batch level variables “MIN TYPES” and “MAX TYPES”	890
Troubleshooting Datacap security and authentication	890
Automatic key import feature fails to apply the new encryption keys	891
Troubleshooting Rulerunner	891
First things to check	892
Viewing Windows Event Viewer logs on Rulerunner server	892
Stopping the Rulerunner Service	893
Enabling Rulerunner processing logs	893
Starting the Rulerunner Service	894
Viewing Rulerunner processing logs	894
Troubleshooting FastDoc	895
Barcode recognition is poor	896
Document type is not automatically assigned	896
Scanner is not listed as an option	897
Extra data included in captured data	897
Index data not picked up automatically in zoned field	897
Correcting export errors	897
SharePoint upload returns file check-in error in Export_rrs.log	898
SharePoint upload returns a list not found error in Export_rrs.log	898
SharePoint upload returns a properties update error in Export_rrs.log	899
FastDoc labels cannot be displayed	899
Troubleshooting Datacap web services	899
Log files	900
Enabling logging for Datacap Desktop	901
Enabling logging for FastDoc	901
Enabling logging for Rulerunner Service	902
Enabling logging for Datacap Server service	903
Enabling logging for the Datacap Web Client	904

Enabling the Datacap Web Services log	904
Best Practices for optimal text recognition	905
Best practices for source control of Datacap applications	905
Parts of a Datacap Application	906
Source Control of Datacap applications	909

Informacje dodatkowe

FastDoc keyboard shortcuts	911
Datacap Web Services REST API methods	914
Session/Logon	916
Session/Logoff	917
Session/ChangeUserPassword	918
Rules/Execute	919
Queue/CreateBatch	922
Queue/DeleteBatches	924
Queue/CheckIntegrity	925
Queue/SaveBatchAttribute	926
Queue/GetBatchAttributes	928
Queue/GetBatchId	929
Queue/GetBatchHistory	930
Queue/GetBatchList	931
Queue/GrabBatch	934
Queue/GrabNextPendingBatchOnJobTaskList	935
Queue/ReleaseBatch	936
Queue/GetCCO	937
Queue/UploadFile	939
Queue/SetFile	940
Queue/GetFile	941
Queue/CopyFilesToCache	942
Queue/SetPageFileName	943
Queue/GetPageFile	944
Queue/GetPageFileName	945
Admin/GetApplicationList	946
Admin/GetProgramFile	947
Admin/SetUserPermissionList	948
Admin/SetGroupPermissionList	949
Admin/GetUserPermissionList	950
Admin/GetGroupPermissionList	952
Admin/SaveTask	953
Admin/GetMobileProfiles	955
Transaction/Start	958
Transaction/SetFile	959
Transaction/Execute	960
Transaction/GetFile	962
Transaction/GetFileList	963
Transaction/End	964
Fingerprint Maintenance Tool reference	965
Fingerprint Maintenance Tool configuration settings	966

Fingerprint Maintenance Tool buttons	966
Fingerprint Maintenance Tool fields	967
Smart Parameter Special Variable Reference	968
Special variables for accessing the application configuration file	969
@APPPATH(<key_path>)	969
@APPVAR(<key_path>)	970
Special variables for accessing the runtime hierarchy	971
@BATCHID	971
@ID	972
@STATUS	972
@VALUE	973
@VAR(<variable_name>)	973
@B.<variable_name>	974
@D.<variable_name>	974
@P.<variable_name>	975
@F.<variable_name>	975
@X.<variable_name>	976
@B\<field_name>[.<variable_name>]	977
@D\<field_name>[.<variable_name>]	978
@P\<field_name>[.<variable_name>]	978
@F\<field_name>[.<variable_name>]	979
@X\<field_name>[.<variable_name>]	979
Special variables for accessing job and task information	980
@JOBID	980
@JOBNAME	981
@OPERATOR	981
@STATION	981
@TASKID	982
@TASKNAME	982
Accessing Parent, Sibling and Child Objects with Smart Parameter Navigation	982
Level Identifier	983
Variable Identifier	983
Type Identifier	984
Miscellaneous special variables	985
@CHR(<unicode_value>)	986
@DATE(<format>)	986
@DCO(<property_name>)	987
@DICT_VALUE(<field>)	987
@DICT_WORD(<field>)	988
@DICT_VINDEX(<csv_string>)	988
@DICT_WINDEX(csv_string)	988
@EMPTY	988
@LOCALE	989
@PATH(<key>)	989
@PILOT(<property_name>)	990
@PROJECTDIR	991
@PROCESSDIR	991
@STRING(<string_value>)	991

@TIME(<format>)	992
@TYPE	992
Smart Parameter Build control	992
Standard Variable Reference	993
Variables that are used on all object types	994
MAX_TYPES	994
MESSAGE	994
MIN_TYPES	995
rules	995
STATUS	996
TYPE	997
hr_locale	997
The hr_SyncImg variable	998
Batch variables	998
LAST_RR_PROFILE	998
ScanSrcChannel	999
Document variables	999
DD	999
Page variables	999
Confidence	1000
DATAFILE	1000
Fingerprint Created	1000
Image_Offset	1001
IMAGEFILE	1001
Latitude	1001
Longitude	1002
PAGE_HEIGHT	1002
PAGE_WIDTH	1002
PageName	1003
PatternConfidence	1003
PD	1003
ScanSrcInputFolder	1004
ScanSrcFileName	1004
ScanSrcSubFolder	1004
ScanSrcPath	1005
TEMPLATE IMAGE	1005
TemplateID	1005
Field variables	1006
DataType	1006
DensityString	1007
DICT	1007
Index	1008
label	1008
Location	1008
Lookup	1009
LookupEx	1010
MaxLength	1010
METRIC	1011

MultiLine	1011
MultiPunch	1012
PatternMatch	1012
PictureString	1012
Pos<templateID>	1014
Position	1014
ReadOnly	1014
RecogStatus	1015
RecogType	1015
ReqConf	1015
SELECT	1016
ShowChar	1016
Sticky	1017
Text	1017
Zone_Offset	1018
Application-specific variable reference	1018
Medical Claims 5010 form configuration parameters	1018
5010 Institutional form configuration variables	1018
5010 Professional form configuration variables	1027
Action library summaries	1037
Global actions	1037
Datacap supported image types	1041
Autodoc actions	1049
BlankPagesIDBySize	1050
CalculateOffset	1051
CreateFingerprint	1051
DeleteFingerprint	1052
FindBlackFingerprint	1053
FindFingerprint	1054
MergeCCOs_ByType	1055
MergeLayoutByType	1055
SetApplicationID	1056
SetFilter_HostName	1057
SetFilter_PageType	1057
SetFingerprint	1058
SetFingerprintDir	1059
SetFingerprintFailureThreshold	1059
SetFingerprintSearchArea	1060
SetFingerprintWebServiceURL	1061
SetMaxOffset	1062
SetProblemValue	1063
UpdateFingerprintStats	1063
Barcode_P actions	1064
Get2DCodeBP	1065
GetAllBarcodesBP	1065
GetBarcodeBP	1066
GetDataMatrixCodeBP	1069
IdentifyByBarcodesBP	1070

MatchBarcodeBP	1071
MatchBarcodePrefixBP	1072
ReadBarCodeBP	1073
SetMinimumConfidenceBP	1074
ClassifyLayout actions	1074
Feedback	1075
Identify	1076
CC actions	1077
ClassifyCC	1078
ClassifyTextCC	1079
RunDecisionPlanCC	1080
RunDecisionPlanForBlocksCC	1081
RunDecisionPlanForTextCC	1083
SetDecisionPlanCC	1084
SetDecisionPlanFieldsCC	1084
SetKnowledgeBaseCC	1085
SetLanguageCC	1085
SetListenerURLCC	1086
SetProblemValueCC	1087
UpdateKnowledgeBaseCC	1087
Category match variables	1088
Cco2cco actions	1089
NormalizeCCO	1089
SetMaxCharacterHeightAVG	1090
SetMaxCharacterHeightTMM	1091
CheckProcessing actions	1091
ProcessCheck	1093
CheckProcessingBrazil	1098
ProcessCheckBrazil	1098
CheckProcessingCanada	1099
ProcessCheckCanada	1100
CheckProcessingFrance	1100
ProcessCheckFrance	1101
CheckProcessingIndia	1102
ProcessCheckIndia	1102
CheckProcessingUK	1103
ProcessCheckUk	1104
CheckProcessingUs	1105
ProcessCheckUs	1105
CMISClient actions	1106
CMISCreateFolder	1107
CMISCreateFolderCustomType	1108
CMISDeleteFile	1109
CMISDeleteFolder	1110
CMISDoesFileExist	1111
CMISDoesFolderExist	1112
CMISDownloadFile	1113
CMISLogDocumentTypes	1114

CMISLogin	1114
CMISRefreshClientCache	1115
CMISSetDocUploadProperty	1116
CMISSetDocUploadType	1118
CMISSetVersion	1119
CMISUploadFile	1120
CMISUploadPage	1121
ColorToBW actions	1122
C2BW_Convert	1122
C2BW_SetAttributes	1123
Convert actions	1124
Common actions	1125
DeleteSourceImagePages	1126
ExceptionSetFileTypes	1126
ExceptionSetHandler	1127
ExceptionSetVariableName	1128
ExceptionSetTaskCondition	1129
SetNamePattern	1130
SetNamePatternFileCheck	1130
PDF compliance standards	1131
Excel actions	1131
ExcelAutoFitColumns	1132
ExcelAutoFitRows	1133
ExcelOrientationToLandscape	1134
ExcelOrientationToPortrait	1135
ExcelPrintBlankPage	1135
ExcelPrintGridlines	1136
ExcelPrintQuality	1137
ExcelScalingFactor	1138
ExcelShapeMinArea	1138
ExcelTiffCompression	1139
ExcelWorkbookToImage	1140
ExcelWorkbookToImageEx	1141
ExcelWorkbookToPdf	1142
Html actions	1143
HtmlLayout	1143
HtmlPrintQuality	1145
HtmlTiffCompression	1145
HtmlToImage	1146
HtmlToPdf	1147
Images actions	1148
ImageDefaultDPI	1148
ImageFileTypesToConvert	1149
ImageMonoThreshold	1150
ImageMonoType	1151
ImageToTIFF	1152
Outlook actions	1153
OutlookAttachmentTypeIndicator	1153

OutlookMessageToAttachmentOnly	1154
OutlookMessageToImageAndAttachment	1155
OutlookPrintQuality	1156
OutlookTiffCompression	1157
Pdf actions	1158
PDFBitDepth	1158
PDFCompression	1159
PDFConversionMethod	1160
PDFDocumentToImage	1161
PDFGrayscale	1162
PDFHorizontalResolution	1162
PDFQuality	1163
PDFVerticalResolution	1164
PdfFRE actions	1165
PDFFREDocumentToImage	1165
PDFFREReleaseEngine	1168
PDF image compression types	1169
Rtf actions	1170
RtfPrintQuality	1170
RtfTiffCompression	1171
RtfToImage	1172
RtfToPdf	1173
Tiff actions	1174
SplitMultipageTiff	1174
SplitTIFFCompression	1175
Txt actions	1176
TxtFontName	1176
TxtFontSize	1177
TxtPrintQuality	1178
TxtTiffCompression	1178
TxtToImage	1179
TxtToPdf	1180
Word actions	1181
WordDocumentToImage	1181
WordDocumentToPdf	1182
WordMonochromeQuality	1183
WordPrintQuality	1183
WordTiffCompression	1184
Zip actions	1185
ZipOverwrite	1185
ZipPassword	1186
ZipUnPack	1187
DatacapBOX actions	1188
Export object	1188
AddParentDataToPageMetadata	1190
CreateBatchSubfolder	1191
DCOVarsAreMetadata	1191
DocumentsToPDF	1192

FailIfFileExists	1193
FieldsAreMetadata	1194
OverwriteExistingFiles	1194
ProcessChildren	1195
ReplaceMetadata	1196
TargetFolder	1196
Upload	1197
Import object	1198
BackupFolder	1199
Download	1200
ImportAsDocumentType	1201
ImportLimit	1201
LookforExtensions	1202
SourceFolder	1203
Dcclip actions	1203
dci_clipfield	1204
DCImageFix actions	1205
ImageEnhance	1205
LoadSettings	1206
LoadSettings_FingerprintID	1207
DCO actions	1208
ChkConfidence	1209
ChkDCOStatus	1210
ChkDCOType	1211
ChkLastDCOTypeEx	1211
ChkIntegrity	1212
ChkLastDCOType	1213
ClearAltText	1214
ClearDCO	1215
CopyPD2DD	1215
CountPagesToDocumentVar	1216
CreateDocuments	1216
CreateFields	1217
DeleteFields	1218
IsDocumentCountMoreThan	1219
IsFirstDocumentInBatch	1219
JoinPreviousDocument	1220
PropagateToAltText	1221
RemoveDocumentStructure	1221
SaveImageInformation	1222
SetDCOStatus	1224
SetDCOType	1225
SetDocStatus	1226
SetDocumentType	1226
SetFldConfidence	1227
SetPageFingerprintID	1228
SetPageStatus	1229
SetPageTemplateID	1230

SetPageType	1230
dcpdf actions	1231
dcpdf_CreateTiffFromPDF	1232
dcpdf_CreateTiffFromPDF_CreateDocs	1233
dcpdf_MakePDFDoc	1235
dcpdf_MaxSizeToReconvert	1236
dcpdf_SetApplication	1237
dcpdf_SetAuthor	1238
dcpdf_SetImageBitcount	1239
dcpdf_SetImageCompression	1240
dcpdf_SetImageGrayscale	1240
dcpdf_SetImageQuality	1241
dcpdf_SetImageResolution	1242
dcpdf_SetKeywords	1243
dcpdf_SetProducer	1243
dcpdf_SetSubject	1244
dcpdf_SetTitle	1245
dcpdf_UseAltConversionMethod	1246
DocumentAnalytics actions	1246
FindLabelValuePairs	1248
FindLabelValuePair	1249
CopyLabelValuePairs	1249
CreateHTML	1250
CopyAllBlocks	1251
AnalyzeLayout	1252
FindPatterns	1253
ExtractText	1255
ExtractTextAlchemyLanguage (deprecated)	1257
ExtractTextLogEnable	1259
ExtractTextNLP	1259
FindExtractedText	1261
DocumentAnalytics.VisualRecognitionClassifier actions	1263
VisualRecogClassify	1263
VisualRecogSetCredentials	1264
VisualRecogSetMinConfidence	1265
VisualRecogSetURL	1266
VisualRecogTrain	1266
DocumentAnalytics.NaturalLanguageClassifier actions	1267
NLCClassify	1268
NLCClassifyText	1269
NLCSetCredentials	1269
NLCSetLanguage	1270
NLCSetMinConfidence	1271
NLCTrain	1272
Documentum actions	1273
DM_Logon	1273
DM_SetContentType	1275
DM_SetFolderName	1275

DM_SetObjectName	1276
DM_UploadDocument	1277
DM_UploadPage	1278
Email actions	1279
SendEMail	1279
SetAttachment	1280
SetBlindCarbonCopyRcpts	1281
SetCarbonCopyRcpts	1281
SetEmailBody	1282
SetMailServer	1283
SetRecipients	1284
SetSender	1284
SetSubject	1285
Equalize actions	1285
EqualizeUnbalancedImage	1286
Ewsmail actions	1287
ex_abort_time	1287
ex_done_folder	1288
ex_EMLOption	1289
ex_ews_version	1290
ex_HTTP_timeout	1291
ex_load_properties_option	1292
ex_login	1293
ex_logout	1294
ex_max_docs	1295
ex_problem_folder	1296
ex_scan	1297
ex_types	1298
ex_wait_time	1299
Export actions	1300
BatchVariable_ExportValue	1302
BlankFields	1303
BlankLines	1303
BPilot	1304
CloseExportFile	1305
DCOProperty	1306
DocumentVariable_ExportValue	1306
ExportAllFields	1307
ExportFieldValue	1308
ExportMYValue	1308
ExportSmartParameter	1309
ExportToBatchDir	1310
Filler	1310
FixedLenLJ	1311
FixedLenRJ	1312
GetDATE	1312
GetProfileString	1313
GetTime	1314

LineItem_AddElement	1315
LineItem_BlankFields	1316
LineItem_ClearElements	1317
LineItem_ExportElements	1317
LineItem_SmartParameter	1318
NewLine	1319
PageVariable_ExportValue	1320
ResetFieldVariables	1320
SaveFilePathAsVariable	1321
SetCSV	1321
SetElementSeparator	1322
SetExportFileEncodingAsASCII	1323
SetExportPath	1324
SetExtensionName	1325
SetFileName	1325
SetFill	1326
SetFixedLength	1327
SetIgnoreFieldStatus	1327
SetJustified	1328
SetOMR_Separator	1329
SetSpaceFill	1329
SetZeroFill	1330
Text	1331
Variable_ExportValue	1331
Variable_IsValue	1332
ExportDB actions	1333
AddRecord	1333
ExportBatchIDToColumn	1334
ExportCloseConnection	1335
ExportFieldToColumn	1336
ExportNodeXMLToColumn	1337
ExportOpenConnection	1338
ExportPropertyToColumn	1339
ExportSmartParamToColumn	1340
ExportToColumn	1341
SetTableName	1343
ExportXML actions	1344
xml_CommitNode	1344
xml_NewNode	1345
xml_SaveFile	1346
xml_SetAttributeValue	1346
xml_SetExportPath	1347
xml_SetFileEncodingAsASCII	1348
xml_SetFileName	1349
xml_SetNodeValue	1349
FileIO actions	1350
CheckFreeDiskSpace	1351
CopyDirectory	1352

CopyFile	1353
DeleteDirectory	1354
DeleteFile	1355
GetFileSize	1356
GetProfileString	1357
IsDirectoryPresent	1358
IsFilePresent	1359
IsFileReadOnly	1360
IsProfilePresent	1361
Readtextfile	1362
RenameFile	1363
SetFileReadOnly	1364
SetProfileString	1365
SplitFileName	1366
ZipOcrResults	1367
FileNetIDM actions	1368
AddAllImagesToDocument	1370
AddFileToDocument	1370
AddPDFImageToDocument	1371
AddTIFImageToDocument	1372
CreateFolder	1373
FileNetDB_ADOConnect	1373
FileNETDocID_SaveAsSmartParameter	1374
FileNETDocID_SetValue	1374
GetDocuments	1375
GetTopFolders	1376
IndexProperty_ID_Component	1376
IndexProperty_ID_DateComponent	1377
IndexProperty_ID_Value	1378
IndexProperty_LeftJUSTIFY	1379
IndexProperty_RightJUSTIFY	1380
IndexProperty_SmartParameter	1380
Library_DMA_Initialize	1381
Library_DS_Initialize	1382
Library_IS_Initialize	1382
Library_LogIn	1383
Library_LogOff	1384
NewDocument	1384
SaveDocToFolder	1385
Upload	1386
Upload_SetDelay	1387
Upload_SetNumAttempts	1387
UseIndexes_OFF	1388
UseIndexes_ON	1389
FileNet P8 actions	1389
FNP8_AddRedactionsToP8Document	1391
FNP8_CreateFolder	1392
FNP8_Login	1392

FNP8_MultiPageDocs	1393
FNP8_SearchAndDownload	1394
FNP8_SetSearchAndDownloadStatusProperty	1396
FNP8_SetSearchClass	1397
FNP8_SetSearchCurrentVersionOnly	1397
FNP8_SetSearchDownloadDir	1398
FNP8_SetSearchFolderRestriction	1399
FNP8_SetSearchIncludeDocsWithoutContent	1400
FNP8_SetSearchIncludeSubClasses	1401
FNP8_SetSearchMaxItems	1402
FNP8_SetSearchOrderBy	1402
FNP8_SetSearchWhereClause	1403
FNP8_SetDefineSecurityParentage	1404
FNP8_SetDestinationFolder	1405
FNP8_SetDocClassId	1406
FNP8_SetDocTitle	1406
FNP8_SetFileMimeType	1407
FNP8_SetFileType	1409
FNP8_SetKeyProperty	1410
FNP8_SetLocale	1411
FNP8_SetMultiValueProperty	1411
FNP8_SetProperty	1412
FNP8_SetPropertyEx	1413
FNP8_SetRetry	1414
FNP8_SetTargetClassID	1414
FNP8_SetTargetObjectID	1415
FNP8_SetTimeout	1416
FNP8_SetUploadMode	1416
FNP8_SetURL	1417
FNP8_UpdateProperties	1418
FNP8_Upload	1418
FNP8_UploadDir	1419
FingerprintMaintenance actions	1420
CloseDatabase	1420
DeleteFingerprint	1421
DeleteFingerprints	1422
OpenDatabase	1423
SetFingerprintFolder	1424
FPXML actions	1424
ReadZonesFPX	1425
SetDetailsAndLineitemPairFPX	1426
SetDirectoryFPX	1427
WriteZoneFPX	1428
WriteZonesFPX	1429
Grayscale actions	1430
ConvertGraytoBW	1430
HandwritingRecognition actions	1431
Recognize	1432

SetAddressApartmentZone	1433
SetAddressCityStateZipZone	1434
SetAddressCityStateZone	1435
SetAddressCityZone	1435
SetAddressStateZone	1436
SetAddressZipZone	1437
SetAmountFormat	1438
SetBoxRemovalMode	1439
SetCreditCardFormat	1440
SetDateFormat	1441
SetDeskew	1442
SetFieldType	1443
SetFullNameFormat	1449
SetLanguage	1450
SetLineRemovalMode	1451
SetMultiLineMode	1452
SetNoiseRemoval	1453
SetPostalDatabase	1454
SetProcessingMode	1455
SetSpecialCharacterSet	1456
SetTemplate	1457
SetValidLength	1458
SetValidValues	1458
SetVocabulary	1459
SetWritingStyle	1460
IBMCM actions	1462
IBMCM_AddPages	1463
IBMCM_CreateFolder	1464
IBMCM_CreateItem	1465
IBMCM_SetAttributeValue	1466
IBMCM_CreateChildItem	1468
IBMCM_SetChildAttributeValue	1468
IBMCM_DeletePages	1470
IBMCM_Logon	1471
IBMCM_ReplacePage	1472
IBMCM_SearchItem	1472
IBMCM_SetSearchOnlyFolderItems	1473
IBMCM_SetMimeType	1474
IBMCM_SetDestinationFolder	1476
IBMCM_SetDestinationFolderEx	1477
IBMCM_StoreItemIDinDCO	1478
IBMCM_UploadDCO_DOC	1479
IBMCM_UploadDCO_Page	1480
IBMCM_SearchAndDownload	1481
IBMCM_SetSearchAndDownloadCriteria	1482
IBMCM_SetSearchAndDownloadDirectory	1483
IBMCM_SetSearchAndDownloadMaximum	1485
IBMCM_SetSearchAndDownloadSort	1486

IBMCM_SetSearchAndDownloadStatusAttribute	1487
ICR_C actions	1488
EnableLoggingICR_C	1489
RecognizeFieldICR_C	1489
RecognizeFieldVoteICR_C	1490
RecognizePageFields2CCO_ICR_C	1491
RecognizePageFieldsICR_C	1492
RecognizePageFieldsICR_CEx	1493
RecognizePageICR_C	1493
RecognizePageToPDFICR_C	1494
ImageConvert actions	1495
AppendAllImages	1496
AppendAllImages_ByType	1496
AppendImage	1497
AppendImage_StartAsNew	1498
ConvertToJPEG	1499
ConvertToTIFF	1500
RescaleImage	1500
SetChrominanceFactor	1501
SetDeleteOriginal	1502
SetGrayScale	1503
SetImageDPIByWidth	1503
SetLuminanceFactor	1504
SetTIFFCompression	1505
ImageFix actions	1506
Imail actions	1506
im_abort_time	1507
im_AcceptMixedAttachments	1508
im_AcceptNoAttachments	1509
im_done_folder	1510
im_login	1511
im_logout	1512
im_max_docs	1513
im_problem_folder	1513
im_reject_types	1514
im_scan	1515
im_SetProxy	1517
im_SortByDate	1518
im_StoreEML	1519
im_types	1520
im_UseSSL	1521
im_wait_time	1521
Imprint actions	1522
AnnotateImage	1523
RedactByRegEx	1524
RedactParameters	1525
SetAdjustedWidth	1526
SetFontName	1527

SetFontSize	1527
SetOpaque	1528
Intellocate actions	1529
iloc_AdjustZones	1529
iloc_AssignPageType	1530
iloc_SetDetailZones	1531
iloc_SetZones	1531
IsPageDataMissing	1532
Invoice actions	1533
AddToDetailErrorMsg	1535
AddToErrorMsg	1535
AllMixedCase	1536
AlterDatebyDay	1536
CalculateNotesZone	1537
CaptureOpInfo	1538
CheckAndFixDecimal	1538
CheckForSticky	1539
CheckFreeDiskSpace	1540
ClearErrorMsg	1541
CreateFingerprint	1541
DetailFix	1542
FindExportImage	1543
FPXMLUsed	1543
GenerateDetails	1544
iloc_SetDetailSimple	1544
IncrementBatchVar	1545
IsFingerPrintClass	1546
IsInINI	1546
IsInList	1547
IsMultipageDocument	1548
IsSinglePageDocument	1548
IsStationIDSuffix	1549
Is_JobNamePrefix	1549
LoadCCOFromField	1550
PopulateZNLineItemFieldDynamic	1551
ReadFPXMLZones	1551
ScanLineItemDynamic	1552
SendOutlookNotification	1553
SetDynamicDetailZones	1553
SetStickyNo	1554
SwapImages	1555
SwitchMMDD	1555
UpdateFPStats	1556
ValidateVendor	1556
WriteErrorMessage	1557
IOverlay actions	1558
Overlay	1558
SetBackgroundImage	1559

SetDitheringBackground	1560
SetHaloBackground	1560
Locate actions	1561
AddKeyList	1565
AggregateKeyList	1566
CreateVirtualPage	1567
CreateVirtualZone	1568
DefaultValue	1570
FilterIt	1570
FindDBList	1571
FindDBList_InZone	1572
FindKeyList	1573
FindKeyList_InZone	1574
FindLastKeyList	1575
FindLastKeyList_InZone	1576
FindLastRegEx	1577
FindLastRegEx_InZone	1577
FindLastRegExList	1578
FindLastRegExList_InZone	1579
FindLastWord	1580
FindLastWord_InZone	1581
FindNextDBList	1582
FindNextDBList_InZone	1583
FindNextKeyList	1584
FindNextKeyList_InZone	1585
FindNextRegExList	1586
FindNextRegExList_InZone	1587
FindRegExList	1588
FindRegExList_InZone	1589
GetSelectedBlockType	1590
GoAboveWord	1591
GoBelowWord	1592
GoDownLine	1593
GoFirstLine	1593
GoFirstWord	1594
GoLastLine	1595
GoLastWord	1596
GoLeftWord	1596
GoRightWord	1597
GoSiblingBlockNext	1598
GoSiblingBlockPrevious	1599
GoUpLine	1600
GroupWords	1601
GroupWordsLEFT	1602
GroupWordsRIGHT	1602
IsAlpha	1603
IsCurrency	1604
IsDateValue	1605

IsNumber	1606
IsSelectedBlockType	1607
IsValue	1608
IsValue_RegEx	1609
LocatePositionRestore	1609
LocatePositionSave	1610
MaxLength	1611
MergeWordLF	1612
MergeWordRT	1613
MinLength	1614
RegExFind	1614
RegExFind_InBlock	1615
RegExFind_InZone	1616
RegExFindNext	1617
RegExFindNext_InBlock	1618
RegExFindNext_InZone	1619
ScanRT	1620
SelectParentBlock	1620
SelectParentBlockOuterType	1622
SelectParentBlockType	1623
SelectSnippet	1624
SetKeyFileEncodingAsUnicode	1625
SetRect	1625
SetVirtualPageEndPosition	1626
SetVirtualPageStartPosition	1627
UpdateDCOField	1628
UpdateField	1628
UpdateFieldWithBlock	1629
ValueInField	1630
ValueInField_Fuzzy	1631
ValueInField_RegEx	1631
WordFind	1632
WordFind_InZone	1633
WordFindNext	1634
WordFindNext_InZone	1635
WordFind_Offset	1636
Lookup actions	1637
ClearLookupResults	1637
CloseConnection	1638
ExecuteSQL	1638
ExecuteSQLEx	1639
OpenConnection	1640
PopulateWithResult	1641
SmartSQL	1642
SmartSQLEx	1643
MC_Identify	1644
AutoField	1644
FindFields	1645

ReadDCOSetup	1646
ReadPageSetup	1647
SetFormType	1648
SetMaxTolerantDistance	1648
MC_Validation	1649
AddCenturyTo2YearDigit	1650
AddToDetailErrorMsg	1651
AddToErrorMsg	1652
CalculateHCFALineCharges	1652
CalculateUBLLineCharges	1653
CheckDocID	1654
ClearErrorMsg	1654
CommonParseAddress	1655
CommonValAddress	1656
ConvertHyphen	1657
FilterPID	1657
FormatFieldLengths	1658
InheritSnippets	1659
MC_ReadZones	1659
Parse31aPhSig	1660
Parse58ainsnm	1661
Parse58binsnm	1661
Parse58cinsnm	1662
ParseConditionCodes	1663
ParseEPSDT	1663
ParseLastFirstIniNames	1664
ParseNDC	1665
PopulateFromField	1666
SetConf	1666
SetOriginalTIF	1667
StripTrailingAlpha	1668
TransformLI	1668
UpdateCredentialList	1670
ValidateNPI	1671
ValProcedureCode	1671
ValRequiredGroup	1672
mvscan actions	1673
mv_retain_folder	1674
scan	1675
set_abort_time	1676
set_copy_folder	1677
set_delete_empty_folders	1678
set_folder	1679
set_image_validation	1680
set_max_docs	1681
set_metadata_types	1682
set_min_age	1683
set_move_wait_time	1684

set_multipage_burst	1685
set_problem_folder	1686
set_sort_method	1687
set_tree_mode	1688
set_types	1688
set_wait_time	1689
Maintenance Manager actions	1690
Application setup actions	1691
SetAdminDB	1691
SetApplication	1692
SetEngineDB	1693
SetPassword	1694
SetServer	1695
SetStation	1696
SetupDisconnectAll	1697
SetupOpenApplication	1697
SetupOpenApplicationEx	1698
SetUser	1700
Query setup actions	1701
QueryClear	1702
QuerySetAge	1702
QuerySetBatchRange	1704
QuerySetBranch	1704
QuerySetDateFormat	1705
QuerySetDateRange	1707
QuerySetDateTimeFormat	1708
QuerySetGeneric	1711
QuerySetJobID	1711
QuerySetOperator	1712
QuerySetPriority	1713
QuerySetSeparator	1714
QuerySetStation	1715
QuerySetStatus	1716
QuerySetTaskID	1717
Batch processing actions	1718
ProcessChangeBatchStatus	1719
ProcessChangeBatchStatusOrder	1720
ProcessChangeBatchStatusTaskOrder	1720
ProcessClearAuditTable	1721
ProcessClearDebugTable	1722
ProcessDeleteBatchesEx	1723
ProcessInjectBatches	1724
ProcessMoveBatchesEx	1725
ProcessMoveDBRecords	1726
ProcessResetPendingOrNotify	1728
ProcessRunSqlQueryEx	1729
Logging actions	1730
LogClear	1731

LogConfigure	1731
LogSendEmail	1732
LogWriteEventLog	1734
LogWriteRecordSet	1735
LogWriteSQLQuery	1735
Reporting actions	1736
ReportQueryTMUsage	1737
ReportSetReportingTable	1737
ReportSetUsageDBTable	1738
OCR_J actions	1739
InitializeEngine	1740
Recognize	1740
ReleaseEngine	1742
OCR_A actions	1742
EnableEngineLogsOCR_A	1743
OCRA_ConvertImage2BW	1744
Recognize	1744
RecognizeBarcodeOCR_A	1751
RecognizeFieldOCR_A	1751
RecognizeFieldVoteOCR_A	1752
RecognizePageFieldsOCR_A	1753
RecognizePageOCR_A	1754
RecognizeToALTOOCR_A	1754
RecognizeToPDFOCR_A	1756
ReleaseEngineOCR_A	1758
RotateImageOCR_A	1759
SetAutoRotationOCR_A	1760
SetConfCalculationParamsOCR_A	1760
SetFastModeOCR_A	1761
OCR_N actions	1762
RecognizePageFieldsOCR_N	1762
RecognizePageOCR_N	1763
OCR_SR actions	1764
Recognize	1765
RecognizeFieldOCR_S	1767
RecognizeFieldVoteOCR_S	1767
RecognizePageFieldsOCR_S	1768
RecognizePageOCR_S	1769
RecognizeToFileOCR_S	1771
RecognizeToPDFOCR_S	1773
RotateImageExOCR_S	1775
RotateImageOCR_S	1776
SetContinueOnFailureOCR_S	1777
SetEngineTimeoutOCR_S	1778
SetOutOfProcessLoggingOCR_S	1779
SetOutOfProcessTimeoutOCR_S	1780
UseOutOfProcessRecogOCR_S	1781
OpenTextFaxServer actions	1781

Connect	1782
ContinueOnConnectionError	1783
ContinueOnFaxImportError	1784
Disconnect	1785
ImportFaxes	1786
SendAsFax	1787
SetAbortTimeout	1788
SetFaxRemovalAfterImport	1789
SetInputFolder	1790
SetMaxNumberOfFaxes	1790
SetNumberOfRetries	1791
SetPollingInterval	1792
SetProcessedFaxesFolder	1793
SetProtocol	1794
SetRetryTimeout	1795
SetServerName	1796
SetUserID	1797
SetUserPassword	1798
SetWindowsAuthentication	1799
PatternMatch actions	1799
MatchPattern	1800
pat_RecogMatch_Id	1801
pat_RegisterZones	1802
pat_ReleasePageAnchors	1803
PatternMatch_Fingerprint	1804
PatternMatch_Identify	1805
PatternMatch_PageType	1806
SetMatchConfidence	1807
Picture actions	1808
PIC_ApplyPictureString	1808
PIC_FilterFields	1809
PIC_FormatFields	1810
PIC_ReplaceBlankField	1812
PIC_SetPictureCharacter	1813
PIC_ValidateField	1814
POLR actions	1814
CallPOLR	1815
Recog_Shared actions	1816
AnalyzeImage	1817
CCONormalization_OFF	1818
CreateTextFile	1819
IsBlankPage	1820
RecogContinueOnFailure	1820
RecogOMRThreshold	1822
RegisterPageFields	1823
RotateTio	1824
SetAdjustFieldToChars	1825
SetFingerprintRecogPriority	1826

SetFullPageRecogArea	1826
SetOutOfProcessRecogTimeout	1827
SetRecogFailureRetryDelay	1828
SetupAutomaticRetry	1829
SnapCCOtoDCO	1831
SnapDCOtoCCO	1832
SnapFieldtoChars	1833
UseOutOfProcessRecog	1833
runner actions	1834
AbortOnError	1836
CheckAllIntegrity	1837
CheckDocCount	1837
CheckPageCount	1838
DebugMode_OFF	1839
DebugMode_ON	1839
GoToNextFunction	1840
MessageID	1841
MessageIDParameter	1841
PilotMessage_Clear	1842
PilotMessage_Set	1843
ProcessChildren	1843
rr_AbortBatch	1844
rr_Get	1844
rr_WriteNode	1845
rrAppend	1846
rrCompare	1847
rrCompareCase	1848
rrCompareCaseLength	1849
rrCompareNot	1850
rrCompareNotCase	1851
rrCompareNotCaseLength	1852
rrCompareNumeric	1853
rrContains	1854
rrCopy	1855
rrPrepend	1856
rrSet	1857
SetBatchPriority	1858
SetOperatorID	1859
SetReturnValue	1859
SetStationID	1860
SetTaskStatus	1861
SkipChildren	1862
Status_Preserve_OFF	1862
Status_Preserve_ON	1863
Task_NumberOfSplits	1863
Task_RaiseCondition	1864
SharedRecognitionTools actions	1865
CreateCcoFromLayout	1865

SignatureValidation actions	1866
CreateNew	1866
SetMinimumConfidence	1867
SetSignatureReferenceFolderPath	1868
ValidateSignature	1869
SPEXport actions	1870
SP_CreateFolder	1871
SP_Login	1872
SP_SetContentType	1872
SP_SetFileType	1873
SP_SetProperty	1874
SP_SetUploadMode	1875
SP_SetUrl	1875
SP_Upload	1876
SP_UploadDir	1877
Split actions	1877
SplitBatch	1878
Statistics actions	1879
AddToDBTotals	1880
CompareFieldsText	1880
IsBatchAborted	1881
SaveFieldsText	1881
TifMerge actions	1882
TifMerge_CheckStatus	1883
TifMerge_ExportToBatchDir	1884
TifMerge_MergeImages	1884
TifMerge_MyImage	1885
TifMerge_PreserveCompression	1886
TifMerge_SetFileName	1887
TifMerge_SetFilePath	1888
TM524 actions	1888
Validations actions	1888
AddLeadingZeros	1892
AddPaddingToEnd	1893
AddPaddingToStart	1893
AddTrailingZeros	1894
AllowOnlyChars	1895
AppendFromField	1895
AppendToField	1896
AssignFieldDefault	1896
CalculateDateDifference	1897
CalculateFields	1898
CheckSubFields	1900
CompareFields	1901
ConvertFieldToCurrency	1902
ConvertToLowerCase	1903
ConvertToUpperCase	1903
CopyField	1904

CopyFieldToField	1904
DateStampField	1905
DeleteAllAlpha	1906
DeleteAllMiscChars	1906
DeleteAllNumeric	1907
DeleteAllPunct	1908
DeleteAllSysChars	1908
DeleteChildType	1909
DeleteLCSpaces	1909
DeleteParentObj	1910
DeleteSelectedChars	1911
EmptyFieldValue	1912
FieldContainsValue	1912
FilterFieldSelectedChars	1913
FormatNumberToLocale	1913
GetJobID	1915
HasChildOfType	1915
InsertChars	1916
InsertDecimalPoint	1917
IsFieldCurrency	1917
IsFieldDate	1918
IsFieldDateEqualOrAfter	1919
IsFieldDateEqualOrBefore	1919
IsFieldDateUpToToday	1920
IsFieldDateWithinRange	1920
IsFieldDateWithinXDays	1921
IsFieldDateWithReformat	1922
IsFieldEmpty	1922
IsFieldFilled	1923
IsFieldGreaterOrEqual	1924
IsFieldHidden	1925
IsFieldLengthMax	1925
IsFieldLengthMin	1926
IsFieldLessOrEqual	1926
IsFieldMatching	1927
IsFieldPercentAlpha	1928
IsFieldPercentNonNumeric	1928
IsFieldPercentNumeric	1929
IsMatchingJobID	1930
IsMaxOMRChecked	1930
IsMinOMRChecked	1931
IsPatternInField	1932
IsSupportedImageFile	1932
IsThisFieldEmpty	1933
IsThisFieldFilled	1934
IsVariableEmpty	1934
IsVariableFilled	1935
ParseMultilineAddress	1936

ParseName	1936
ReplaceChars	1937
ReplaceValueAtPosition	1938
ResetField	1938
SetIsOverrideable	1939
SplitFieldValuePreserveEnd	1940
SplitFieldValuePreserveStart	1940
SumFields	1941
TimeStampField	1942
TrimSpaces	1943
TruncateFromEnd	1943
TruncateFromStart	1944
Vote actions	1944
VoteFld	1945
Vscan actions	1945
AddDocument	1946
CopyFile	1947
DeleteImageFile	1948
MoveImageFileToDirectory	1949
Scan	1950
SearchInSubdirectory	1951
SetAlternateImageNames	1951
SetFastMode	1952
SetImageType	1953
SetMaxImageFiles	1954
SetMultiPageTiff	1954
SetSortOrder	1955
SetSourceDirectory	1956
Web Services actions	1957
WsClearHeaders	1958
WsClearParameters	1958
WsClearResultItems	1959
WsEncodeParameter	1960
WsGetFile	1961
WsGetValues	1961
WsSetCredentials	1962
WsSetHeader	1963
WsSetNamespace	1964
WsSetParameter	1964
WsSetResultItem	1965
WsSetTimeout	1966
WsUploadData	1966
WsUploadFile	1967
Zones actions	1968
AdjustZonesToImageOffset	1970
AnchorPage	1971
CalculateLocalOffset	1971
CreateBlockCCO	1972

FindBlocks_WhiteSpace	1973
FindDataBlocks	1973
FindRegExBlocks	1974
FindZoneLineItems	1975
GetZoneText	1976
InheritParentPosition	1976
LoadBlockCCO	1977
LoadZones	1978
MCCOPositionAdjust	1978
MergeZones	1979
PadZone	1979
PopulateZNField	1980
PopulateZNLineItemField	1981
ReadZones	1981
RegisterPage	1982
ScanDetails	1983
ScanDetailsByLines	1983
ScanDetailsByVSpace	1984
ScanLineItem	1985
SetEOL	1985
SetEOL_CRLF	1986
ZoneBOTTOM_ImageBottom	1987
ZoneBOTTOM_LowerBound	1987
ZoneBOTTOM_UpperBound	1988
ZoneImage_SaveAs	1989
ZoneLEFT_ImageLeft	1989
ZoneLEFT_LeftBound	1990
ZoneLEFT_RightBound	1991
ZoneRIGHT_ImageRight	1991
ZoneRIGHT_LeftBound	1992
ZoneRIGHT_RightBound	1993
ZoneTOP_ImageTop	1993
ZoneTOP_LowerBound	1994
ZoneTOP_UpperBound	1995
Application specific actions	1995
Medical Claims actions	1995
4010Common	1996
4010Institutional	1996
4010Professional	1996
5010Common	1996
5010Institutional	1996
5010Professional	1997
MC_Validation	1997
Datacap Accounts Payable actions	1998
APT_Localization	1999
APTCustom	2000
ConcatLineValues	2000
Documents	2001

FlexID	2001
Intellocate_Learning	2001
PageID	2002
PreVerifySetup	2002
Redaction	2002
Adding keyboard shortcuts to an application	2003
Keyboard shortcuts for ASPX web pages	2003
Keyboard shortcuts for the Datacap Desktop Scan task	2005
Keyboard shortcuts for the Datacap Desktop Fixup task	2006
Keyboard shortcuts for the Datacap Desktop Verify task	2007

Indeks 2008

Dokumentacja programu IBM Datacap 9.1.3

Witamy w dokumentacji programu IBM® Datacap w wersji 9.1.3, w której znajdziesz informacje dotyczące instalacji, obsługi i używania programu IBM Datacap.

Pierwsze kroki

- [Co nowego](#)
- [Przegląd](#)
- [Planowanie systemu Datacap](#)
- [Pierwsze kroki w tworzeniu aplikacji](#)
- [Kokpit Datacap Navigator](#)
- [Dokumentacja rozwiązania Medical Claims](#)
- [Scenariusz: Rozwiązanie z zakresu przetwarzania obrazów ułatwiające przechowywanie dokumentów](#)
- [Scenariusz: Automatyzacja przetwarzania zamówień dla aptek wysyłkowych](#)
- [🔗 Uwagi do wydania](#)
- [Wymagania sprzętowe i programowe dotyczące IBM Datacap](#)

Typowe czynności

- [Instalowanie programu IBM Datacap na pojedynczym serwerze](#)
- [Instalowanie i konfigurowanie programu IBM Datacap w środowisku klient/serwer](#)
- [Migracja z innych wersji](#)
- [Tworzenie aplikacji](#)

Rozwiązywanie problemów i wsparcie

- [Rozwiązywanie problemów z programem Datacap](#)
- [🔗 Portal wsparcia dla programu IBM Datacap](#)
- [🔗 Strona główna działu wsparcia IBM dla oprogramowania](#)

Więcej informacji

- [🔗 Dokumentacja aplikacji Datacap Mobile dla systemu iOS](#)
- [🔗 Dokumentacja aplikacji Datacap Mobile dla systemu Android](#)
- [🔗 Szkolenia i certyfikaty](#)
- [🔗 IBM Datacap 9.0 – pakiet programisty](#)
- [🔗 Centrum aplikacji IBM do zarządzania treścią w przedsiębiorstwie](#)
- [🔗 Globalna społeczność użytkowników rozwiązań do zarządzania treścią w przedsiębiorstwie od IBM](#)
- [🔗 Rozwiązania IBM do zarządzania treścią w przedsiębiorstwie w LinkedIn](#)
- [🔗 Rozwiązania IBM do zarządzania treścią w przedsiębiorstwie na Twitterze](#)

© Copyright IBM Corp. 2017

Co nowego w wersji 9.1.3 programu IBM Datacap

Sekcja „Co nowego” zawiera informacje o nowych funkcjach i zmianach wprowadzonych w najnowszej wersji IBM® Datacap 9.1.3 względem poprzedniej wersji IBM Datacap 9.1.2.

- [Co nowego w wersji 9.1.3 programu IBM Datacap](#)
Wersja 9.1.3 programu IBM Datacap oferuje nowe funkcje i udoskonalenia.

Co nowego w wersji 9.1.3 programu IBM Datacap

Wersja 9.1.3 programu IBM® Datacap oferuje nowe funkcje i udoskonalenia.

Udoskonalenia w programie Datacap Navigator

- Monitor zadań programu Datacap Navigator – wyszukiwanie wg daty

W Monitorze zadań programu Datacap Navigator można wyszukiwać partie według daty. Wyszukiwanie nie ogranicza się do istniejących załadowanych stron. W zapytaniach o daty obsługiwane są operatory równości, poprzedzania, następstwa i przedziału. [Więcej informacji...](#)

- Przy skanowaniu transakcyjnym z programu Navigator lub IBM Case Manager obsługiwana jest przeglądarka wirtualna Daeja.
- W programie Datacap Navigator wprowadzono następujące nowe funkcje i udoskonalenia:
 - Speed Scan Client: klient szybkiego skanowania
 - Batch Prep Client: klient przygotowania partii
 - Speed Index Client: klient indeksowania szybkiego
 - Równoległe przesyłanie stron, które zwiększa wydajność przesyłania
 - Obsługa przesyłania asynchronicznego

[Więcej informacji...](#)

Ułatwienia w używaniu

Datacap Navigator oferuje następujące udoskonalenia, dzięki którym możliwe jest:

- zaktualizowanie filtru;
 - skonfigurowanie przycisków w panelu pola klienta Verify;
 - usunięcie jednym kliknięciem wszystkich stron w widoku miniatur. Ta funkcja jest dostępna tylko w kliencie Speed Scan;
 - wyświetlenie listy zapisanych filtrów;
 - zmiana statusu partii z powrotem na Oczekująca;
 - usunięcie wszystkich obrazów w partii jednym kliknięciem w panelu skanowania;
 - wyświetlanie zrozumiałych etykiet w drzewie partii klientów Scan, Fix-up i Verify, po zeskanowaniu dokumentów ze skanera lub folderu lokalnego bądź po skopiowaniu stron lub podzieleniu dokumentów;
 - zmiana statusu partii z powrotem na Oczekujące w klientach, takich jak klient Speed Scan i Speed Index;
 - wyświetlanie paska przewijania w przeglądarce wirtualnej, w widoku miniatur;
 - sortowanie odfiltrowanej listy partii w Monitorze zadań;
 - tworzenie predefiniowanych filtrów (z wieloma kryteriami) i zapisywanie filtrów;
 - stosowanie zapisanych filtrów bez konieczności tworzenia ich za każdym razem od nowa;
 - usuwanie filtrów.
- Możliwe jest używanie adnotatorów ExtractText w przypadku korzystania z tekstu zajmującego więcej niż jeden blok na poziomie dokumentu.

Nowe działania w bibliotece działań programu Datacap

- **Działanie rrCompareNumeric**

Działanie rrCompareNumeric umożliwia porównywanie wartości liczbowych za pomocą operatorów mniejszości i większości. [Więcej informacji...](#)

- **Działanie SaveImageInformation**

Działanie SaveImageInformation uzyskuje informacje o obrazie i zapisuje je w obiekcie DCO. [Więcej informacji...](#)

- **Działanie Visual Recognition Classifier**

Dodano nowe działanie służące do szkolenia klasyfikatora Visual Recognition Classifier. Działanie to jest podobne do działania służącego do uczenia klasyfikatora Natural Language Classifier. Ponadto zmieniono się działanie classify zarówno dla klasyfikatora IBM Watson Natural Language Classifier, jak i VisualAge — obecnie jego parametrem jest nazwa klasyfikatora. Dzięki temu autor aplikacji nie musi używać identyfikatora klasyfikatora, który można traktować jako wewnętrzną wartość +F28. [Więcej informacji...](#)

- **Działanie Natural Language Classifier**

Dodano nowe działania służące do klasyfikacji tekstu przy użyciu klasyfikatora IBM Watson Natural Language Classifier. [Więcej informacji...](#)

- **Nowe działania IBM Content Manager**

Te działania są przydatne do wyszukiwania i pobierania dokumentów programu IBM Content Manager. [Zobacz inne działania](#)

- **Nowe działania FileNet P8**

Te działania są przydatne do wyszukiwania i pobierania dokumentów. [Zobacz inne działania](#)

- **Działanie rrContains**

Działanie rrContains umożliwia manipulowanie częścią łańcucha w zmiennej lub w polu. Na podstawie wprowadzonych parametrów inteligentnych wyszukuje i porównuje wartości zmiennych dwóch obiektów. [Więcej informacji...](#)

- **Działanie CreateVirtualZone**

To działanie tworzy strefę pola w czasie wykonywania na podstawie zidentyfikowanych położeń tekstu. [Więcej informacji...](#)

- **Nowe działanie służące do ustawienia klucza grupy filtrów zestawu partii**

Dodano nowe działanie umożliwiające twórcom aplikacji ustawienie nowego klucza grupy filtrów partii w tabeli tmbatch. Działanie akceptuje jedną lub więcej nazw grup i zapisuje łańcuch lub (co jest preferowane) odwzorowaną wartość całkowitą (obliczaną na podstawie trybu). Po zwolnieniu partii klient (Rulerunner, FastDoc, DcDesktop lub Navigator) przekazuje tę informację do aTM.

- **Działanie RefreshFields**

Działanie RefreshFields tworzy pola, ale zachowuje istniejące dane. Jest podobne do działania CreateFields, z tym że jeśli istnieje plik danych dla strony, to poprzednio utworzone pola nie są usuwane, a do strony dodawane są nowe pola.

- **Działania AddDcoNode**

Działanie AddDcoNode tworzy węzeł DCO typu wskazanego przez pole, stronę lub dokument. Nowe węzły można dodawać na wszystkich poziomach (tj. dokumentu, strony i pola). Jeśli istnieje węzeł tego samego typu o tym samym identyfikatorze, to nowy węzeł nie jest tworzony, a działanie zwraca wartość True.

Działanie `rrSet_ID` zmienia nazwę docelowego węzła DCO tak, by była równa wartości ze źródła. Przypisuje docelowemu identyfikatorowi (`dco.id`) wartość ze źródła. Działanie zwraca `False` tylko wtedy, gdy nie może odszukać obiektu docelowego. W przeciwnym razie zwraca `True`.

Udoskonalenia ogólne

- **Wyłączanie ikony zamknięcia na karcie**

W ustawieniach głównych funkcji programu Datacap na pulpicie dodano ustawienie umożliwiające włączenie lub wyłączenie ikony zamknięcia karty (Monitor zadań, Szybkie skanowanie). Domyślnie ikona zamknięcia jest włączona.

- **Utajnianie wielostronicowych plików i PDF**

Możliwe jest utajnianie wielostronicowych plików TIFF i PDF.

- **Filtrowanie partii na podstawie ról i grup w środowiskach DcDesktop i FastDoc**

Możliwe jest filtrowanie partii na podstawie ról i grup w środowiskach DcDesktop i FastDoc.

- **Masowe pobieranie treści FileNet P8 przy użyciu zadania przeglądania w systemie FileNet.**

Można wykorzystać środowisko przeglądania FileNet do masowego pobierania treści systemu FileNet P8. [Więcej informacji...](#)

- **Optymalizacja wydajności**

Odświeżanie stron interfejsu użytkownika służących do weryfikacji i klasyfikowania trwa teraz krócej, gdy używane jest uwierzytelnianie LLDAP.

- **Ustawienie tabel OCR/A nakazujące rozpoznawanie każdej linijki jako wiersza tabeli**

Udostępniono ustawienie mechanizmu ABBYY związane z rozpoznawaniem tabel. Nakazuje ono rozpoznawanie każdej linijki w tabeli jako osobnego wiersza tej tabeli. Jeśli każda linijka jest osobnym wierszem tabeli, a mimo to mechanizm rozpoznaje komórki wielowierszowe, można użyć tego ustawienia, aby wymusić traktowanie każdej linijki jako osobnego wiersza.

- **Udoskonalenie programu Datacap Application Manager**

Udoskonalony program Datacap Application Manager umożliwia wybranie wartości "{domyślne}" dla niektórych ustawień aplikacji, tak aby obowiązywały ustawienia ze strony serwera. Do listy mechanizmów uwierzytelniania dodano ADLDS.

- **Większa skuteczność identyfikacji dzięki możliwości zdefiniowania strefy w lokalizacji**

Dodano mechanizm udostępniający OCRA współrzędne tabeli dla wybranego zestawu reguł. Służy to usprawnieniu identyfikacji tabel, gdy możliwe jest wyznaczenie strefy w lokalizacji.

- **Wyświetlanie dostępnych szablonów partii dla wybranego zestawu reguł**

Dodano okno dialogowe wyświetlające dostępne szablony partii dla wybranego zestawu reguł. Dodano także przycisk "URUCHOM" inicjujący działanie wybranego zestawu reguł.

- **W programie Datacap Studio dodano kreator parametrów inteligentnych**

W programie Datacap Studio podczas ustawienia parametrów działań dostępny jest nowy kreator parametrów inteligentnych. [Więcej informacji...](#)

Inne nowe możliwości

- **Możliwość skanowania na komputerach Apple Macintosh**

Możliwe jest skanowanie przy użyciu komputera Apple Macintosh.

- **Możliwość zaktualizowania liczb stron i dokumentów po zakończeniu skanowania**

Możliwe jest zaktualizowanie liczb stron i dokumentów wyświetlanych w Monitorze zadań bezpośrednio po zakończeniu czynności skanowania.

- **Możliwość wyświetlania list wyników wyszukiwania usług Web Services**

Listy wyników wyszukiwania usług Web Services można wyświetlać w interfejsie użytkownika służącym do weryfikacji.

- **OCR/A może rozpoznawać tabele wyłącznie na podstawie linii siatki**

OCR/A może rozpoznawać tabele wyłącznie na podstawie linii siatki. Obsługiwane jest ustawienie konfiguracyjne `y_SplitOnlyBySeparators` w bibliotece Abbyy OCR/A, które nakazuje mechanizmowi rozpoznawanie tabel wyłącznie na podstawie linii siatki na stronie.

- **Zmiana mechanizmu rozpoznawania OCR/SR**

Moduł OCR/SR wyposażono w nowy mechanizm rozpoznawania w wersji 20

- **Możliwe jest określanie konfigurowalnych kryteriów ustalania priorytetów dla kolejkowania w programie Datacap**

W programie Datacap obsługiwane jest kolejkowanie na podstawie bliskości "wymaganego terminu". Możliwe jest dodatkowo kolejkowanie według godziny rozpoczęcia zadania i kolejkowanie według godziny rozpoczęcia czynności. Gdy dla partii niestandardowej zostanie określony termin (data/godzina), serwer może prowadzić sortowanie według tej kolumny.

- **Funkcje klasyfikacji obrazów**

Możliwa jest klasyfikacja obrazów przy użyciu usługi Watson Image Recognition.

Komponenty i funkcje nieaktualne

- Z produktu Datacap usunięto bibliotekę działań OCR/S.
- Usunięto opcje dotyczące kodów kreskowych i pisma odręcznego z karty OCR/S w programie Datacap Studio.

Temat nadrzędny: [Co nowego w wersji 9.1.3 programu IBM Datacap](#)

Ułatwienia dostępu w programie Datacap Navigator

Program Datacap Navigator oferuje funkcje ułatwiające korzystanie z niego osobom niepełnosprawnym.

Ważne: Ułatwienia dostępu są obsługiwane tylko w przeglądarkach WWW dla systemów operacyjnych z rodziny Microsoft Windows.

Sterowanie i nawigacja za pomocą klawiatury

Dostępne są następujące funkcje umożliwiające wprowadzanie danych i nawigację za pomocą klawiatury:

Sterowanie programem z klawiatury

Do obsługi programu Datacap Navigator można zamiast myszy używać klawiatury. Aby skorzystać z wybranego przycisku w interfejsie użytkownika, należy przejść do tego przycisku i nacisnąć klawisz Enter. Aby wprowadzić dane, należy przejść do pola wprowadzania danych, wpisać dane i nacisnąć klawisz Enter lub klawisz Tab, który spowoduje wyjście z pola.

Miejsce aktywne dla klawiatury

Miejsce aktywne dla klawiatury jest wyróżnione lub podświetlone. Jest to aktywny obszar okna, do którego kierowane są informacje o naciśniętych klawiszach.

Nawigacja za pomocą klawiatury

Klawisze Tab, Shift+Tab i strzałek umożliwiają przechodzenie między głównymi elementami strony, widoku lub sekcji specjalnej. W przypadku niektórych elementów, takich jak widoki drzewa lub selektory dat, można też używać klawiszy Home, End, Page Up i Page Down. Naciśnięcie klawiszy Ctrl+Shift+strzałka w dół na stronie przeglądania umożliwi nawigację za pomocą banera, paska narzędzi, widoku drzewa i widoku listy.

Możesz użyć narzędzia JAWS w celu pominięcia sekcji treści przez przejście do obszarów punktów orientacyjnych na stronie. W programie JAWS naciśnij klawisze Insert+Ctrl+;, aby otworzyć okno dialogowe zawierające listę punktów orientacyjnych. Wybierz punkt orientacyjny, aby pominąć bloki sekcji i przejść do tego obszaru. Na przykład można wybrać opcję **główny** z listy punktów orientacyjnych w celu przejścia do głównej treści strony. Alternatywnie można przejść do głównej treści strony, naciskając klawisze Insert+F7 w programie JAWS w celu otwarcia okna dialogowego zawierającego listę ukrytych odsyłaczy na stronie. Można wybrać opcję **Przejdź do głównej treści** w celu przejścia do głównej treści, lub wybrać opcję **Przejdź do przycisków nawigacji** w celu przejścia do przycisków nawigacji po lewej stronie. Użyj klawisza Tab, aby przejść do odsyłaczy, naciśnij klawisz Enter, aby wybrać odsyłacz, a następnie naciśnij klawisz Tab, aby uaktywnić obszar docelowy powiązany z wybranym odsyłaczem.

Kliknięcie i przesunięcie obszaru aktywnego za pomocą klawisza Tab

W celu efektywnego postępowania się klawiaturą należy zrozumieć różnice między kliknięciem a przesunięciem obszaru aktywnego za pomocą klawisza Tab. Jeśli użytkownik korzysta tylko z myszy, wówczas klawisz Tab jest nieistotny; jednak korzystanie z niego ma znaczenie w przypadku korzystania ze skrótów klawiszowych.

- Obszar aktywny ustawiony za pomocą klawisza Tab jest oznaczony kropkowanym kwadratem wokół elementu. Ustawienie obszaru aktywnego można zmienić za pomocą klawisza Tab i klawiszy strzałek, albo poprzez kliknięcie elementu przyciskiem myszy.
 - Podczas nawigowania za pomocą klawisza Tab obszar aktywny jest przesuwany kolejno od jednego elementu do następnego, a obszar aktywny wybrany poprzez kliknięcie nie ulega zmianie. Po osiągnięciu ostatniej sekcji strony obszar aktywny przesuwany za pomocą klawisza Tab przenosi się do pierwszego elementu w następnej sekcji.
 - W przypadku korzystania z klawiszy strzałek obszar aktywny przesuwany klawiszami oraz obszar aktywny wybierany kliknięciem przesuwają się kolejno, w tej samej sekcji ekranu.
- Obszar aktywny ustawiony przez kliknięcie jest wyróżniony kolorowym podświetleniem wokół elementu. Aby zmienić obszar aktywny ustawiony za pomocą myszy, należy kliknąć element myszą, użyć klawiszy strzałek albo użyć klawisza Tab, a następnie nacisnąć Enter.

Gdy aplikacja Datacap Navigator otworzy nowe okno, np. przeznaczone do konkretnego działania, albo wyświetli okno dialogowe zawierające np. ostrzeżenia i błędy, obszar aktywny jest ustawiany u góry strony (po zamknięciu okna albo okna dialogowego). Aby wrócić do pierwotnego położenia obszaru aktywnego, użyj kombinacji klawiszy Ctrl+Shift+Strzałka w dół i Strzałka w górę lub Tab.

Instalacja cicha

Aby skorzystać z instalacji z ułatwieniami dostępu, należy wybrać instalację cichą.

Skróty klawiszowe

Dostęp do wszystkich funkcji programu Datacap Navigator jest możliwy za pośrednictwem klawiatury.

Dostęp z wykorzystaniem klawiatury spełnia standardowe wytyczne systemu Microsoft Windows.

Dostęp z klawiatury różni się od standardowych wytycznych obowiązujących w systemie Microsoft Windows w następujących aspektach:

Klawisze dostępu, przechodzenie klawiszem Tab i tabele

Klawisze dostępu są przypisane tylko do przycisków i elementów menu. Do dowolnego pola można natomiast przejść, naciskając klawisz Tab.

Naciśnij klawisz Tab, aby przenieść kursor do tabeli. Ponownie naciśnij klawisz Tab, aby przenieść kursor do następnej komórki w tabeli. Aby wyjść z tabeli do następnego pola, naciśnij i przytrzymaj klawisz Ctrl, a następnie naciśnij klawisz Tab. Gdy kursor znajduje się w tabeli, naciśnięcie klawisza Enter nie jest równoważne kliknięciu przycisku OK w celu zamknięcia okna; najpierw należy wyjść z tabeli.

Aby edytować komórkę tabeli zawierającą pole złożone, naciśnij klawisz F2, za pomocą klawiszy ze strzałkami w górę i w dół przejdź do żądanej pozycji, a następnie naciśnij klawisz Enter, aby ją wybrać.

Karty i stronicowanie

Między kartami można przechodzić za pomocą klawiszy ze strzałką w lewo i w prawo.

W celu przejścia na pierwszą kartę należy nacisnąć klawisz Home. W celu przejścia do ostatniej karty należy nacisnąć klawisz End.

W celu przejścia do następnej strony należy nacisnąć klawisz Ctrl z klawiszem Page Down, lub nacisnąć jednocześnie Ctrl+Tab.

W celu przejścia do poprzedniej strony należy nacisnąć klawisz Ctrl z klawiszem Page Up, lub nacisnąć jednocześnie Ctrl+Shift+Tab.

Pola złożone

Aby przejść do elementu, należy użyć klawiszy strzałek w górę i w dół. Następnie nacisnąć klawisz Enter, aby dokonać wyboru.

Menu

W systemie operacyjnym Windows otwórz menu dla aktywnej pozycji, naciskając kombinację klawiszy Shift+F10.

W celu nawigowania w pozycjach menu można korzystać z klawiszy strzałek w górę i w dół. Następnie należy nacisnąć klawisz Enter lub klawisz spacji, aby aktywować pozycję menu lub aby otworzyć podmenu.

W celu zamknięcia menu lub podmenu należy nacisnąć klawisz Esc lub klawisz strzałki w lewo.

Widoki drzew

Wpisywanie znaków lub naciskanie klawisza Backspace, gdy aktywny jest element drzewa, nie powoduje wybrania elementu.

Jeśli w systemie jest zainstalowany pakiet Java™ 2 Software Development Kit 1.4, to po naciśnięciu klawisza z literą wybrany zostanie następny element w drzewie rozpoczynający się od tej litery.

W celu nawigowania w widokach drzew można korzystać z następujących klawiszy:

Tabela 1. Nawigacja w widoku drzewa

Działanie	Klawisz
Nawigacja do drzewa	Tab

Działanie	Klawisz
Nawigacja do następnego węzła równorzędnego	Strzałka w górę
Otwarcie poddrzewa	Strzałka w prawo
Zamknięcie poddrzewa	Strzałka w lewo
Nawigacja do otwartego poddrzewa	Strzałka w prawo
Nawigacja do węzła nadrzędnego	Strzałka w lewo
Aktywacja elementu drzewa	Enter
Nawigacja do pierwszego węzła drzewa	Home
Nawigacja do ostatniego widocznego węzła drzewa	End

Pola listy, pola wyboru i przyciski opcji

W polu listy przejdź do żądanej pozycji elementu za pomocą klawiszy ze strzałką w górę i w dół, a następnie naciśnij klawisz Enter, aby dokonać wyboru. Aby wybrać kilka pozycji następujących bezpośrednio po sobie, naciskaj klawisz ze strzałką w górę lub w dół, jednocześnie trzymając naciśnięty klawisz Shift.

Jeśli w systemie jest zainstalowany pakiet Java 2 Software Development Kit 1.4, to można wybrać pozycję w polu listy, polu złożonym lub tabeli, naciskając klawisz z odpowiednią literą.

W polach listy można wybierać pojedyncze przełączniki, naciskając klawisz Tab, a następnie klawisz spacji, lub korzystając z klawiszy dostępu. Klawisze ze strzałkami nie wybierają przycisków opcji wewnątrz grupy.

Próbnik dat kalendarzowych

W celu nawigowania wśród komórek dat należy korzystać z klawiszy strzałek w lewo, w prawo, w górę i w dół.

W celu przelączenia na ten sam dzień następnego miesiąca należy nacisnąć klawisz Page Down.

W celu przelączenia na ten sam dzień poprzedniego miesiąca należy nacisnąć klawisz Page Up.

W celu przelączenia na ten sam dzień następnego roku należy nacisnąć klawisze Ctrl+Page Down.

W celu przelączenia na ten sam dzień poprzedniego roku należy nacisnąć klawisze Ctrl+Page Up.

W celu przejścia do pierwszego dnia miesiąca należy użyć klawisza Home.

W celu przejścia do ostatniego dnia miesiąca należy użyć klawisza End.

Naciśnięcie klawisza Enter powoduje wybór daty.

Pomoc w dymkach

Za pomocą klawisza Tab przejdź do pola lub elementu z pomocą w dymku. Następnie naciśnij klawisze Ctrl+F1, aby wywołać pomoc w dymku.

Nawigacja w oknach wywoływanych

W różnych sytuacjach program Datacap Navigator wyświetla informacje lub komunikaty o błędach w oknach wyskakujących. W tej sekcji został wyjaśniony sposób nawigowania za pomocą klawiatury i innych ułatwień dostępu w każdym typie okna.

Tabela 2. Klawisze potrzebne do nawigacji w oknach wyskakujących bez użycia myszy

Typ okna wyskakującego	Opcje klawiatury
Informacja	Aby zamknąć, naciśnij klawisz Esc.
Alert	Aby zamknąć, przejdź klawiszem Tab do przycisków dostępnych w oknie.
Błąd	Aby wyświetlić pełny komunikat o błędzie, przejdź klawiszem Tab do Wyjścia błędów i naciśnij klawisz Enter. Następnie przejdź klawiszem Tab do komunikatu o błędzie. Aby zamknąć, naciśnij klawisz Esc. Wskazówka: W trybie wirtualnego kursora PC obszar aktywny jest ponownie ustawiany u góry bieżącej strony po zamknięciu okna wywoływanego.
Dane wejściowe	Wpisz żądane informacje. Aby zamknąć, przejdź klawiszem Tab do przycisku OK lub Anuluj i naciśnij klawisz Enter.

Zgodność z rozwiązaniami technicznymi dla niepełnosprawnych

Program Datacap Navigator jest zgodny z lektorem ekranowym JAWS. Datacap Navigator ma właściwości, które są wymagane przez aplikację JAWS do udostępnienia wyświetlanych na ekranie informacji użytkownikom z wadami wzroku.

Wymaganie: Lektor ekranowy JAWS musi być uruchomiony za pomocą komendy java, a nie komendy javaw; lektor ekranowy nie będzie działał prawidłowo, jeśli zostanie uruchomiony za pomocą komendy javaw.

Dokumentacja produktu

Dokumentacja tego produktu dostępna jest w formatach uwzględniających ułatwienia dostępu.

Dokumentacja jest dostępna w formacie HTML z ułatwieniami dostępu. Zastosowanie formatu HTML umożliwia wyświetlenie dokumentacji zgodnie z preferencjami wyświetlania określonymi w używanej przeglądarce. Możliwe jest również używanie lektorów ekranowych i innych rozwiązań technicznych dla użytkowników niepełnosprawnych.

Dokumentacja jest także dostępna w formacie PDF.

- [Skróty klawiszowe w programie Datacap Navigator](#)

W programie Datacap Navigator można używać skrótów klawiszowych do nawigacji i wybierania elementów interfejsu użytkownika służących do skanowania, przekazywania, klasyfikacji i weryfikacji.

Temat nadrzędny: [Dokumentacja programu IBM Datacap 9.1.3](#)

Skróty klawiszowe w programie Datacap Navigator

W programie Datacap Navigator można używać skrótów klawiszowych do nawigacji i wybierania elementów interfejsu użytkownika służących do skanowania, przekazywania, klasyfikacji i weryfikacji.

Tabela 1. Skróty klawiszowe dotyczące czynności skanowania w programie Datacap Navigator

Skrót klawiszowy	Funkcja
Ctrl + 1	Wyślij
Ctrl + 2	Wstrzymaj
Ctrl + 4	Anuluj

Skrót klawiszowy	Funkcja
Ctrl + U	Przejdź o jedną stronę w górę w strukturze partii
Ctrl + D	Przejdź o jedną stronę w dół w strukturze partii
Ctrl + F10	Usuń stronę w strukturze partii
Ctrl + F11	Usuń wszystkie strony w strukturze partii

Tabela 2. Skróty klawiszowe dotyczące czynności przekazywania w programie Datacap Navigator

Skrót klawiszowy	Funkcja
Ctrl + 1	Wyślij
Ctrl + 2	Wstrzymaj
Ctrl + 3	Przerwij

Tabela 3. Skróty klawiszowe dotyczące czynności klasyfikacji w programie Datacap Navigator

Skrót klawiszowy	Funkcja
Ctrl + ,	Przejdź o jedną stronę w górę w strukturze partii
Ctrl + .	Przejdź o jedną stronę w dół w strukturze partii
Ctrl + 1	Wyślij
Ctrl + 2	Wstrzymaj
Ctrl + 5	Przejdź do poprzedniej strony w strukturze partii
Ctrl + 6	Przejdź do następnej strony w strukturze partii
Ctrl + 7	Przejdź do poprzedniej strony z błędem w strukturze partii
Ctrl + 8	Przejdź do następnej strony z błędem w strukturze partii
Ctrl + 9	Wyświetl i przełącz superzmienną w tytule przeglądarki obrazów
Ctrl + F1	Rozwiń lub zwiń węzeł, aby uwidocznić lub ukryć węzeł podrzędny w strukturze partii
Ctrl + F2	Skopiuj stronę do bieżącego dokumentu
Ctrl + F3	Podziel dokument w strukturze partii
Ctrl + F8	Połącz wszystkie dokumenty z pierwszym dokumentem
Ctrl + F9	Sprawdź integralność struktury partii
Ctrl + I	Oznacz stronę do ponownego skanowania
Ctrl + Q	Oznacz stronę lub status dokumentu jako Usunięto lub Problem
Ctrl + V	Oznacz stronę lub dokument do recenzji
Ctrl + Y	Dołącz bieżący dokument do poprzedniego

Tabela 4. Skróty klawiszowe dotyczące czynności weryfikacji w programie Datacap

Navigator

Skrót klawiszowy	Funkcja
Ctrl + 1	Wyślij
Ctrl + 2	Wstrzymaj
Ctrl + 5	Przejdź do poprzedniej strony w strukturze partii
Ctrl + 6	Przejdź do następnej strony w strukturze partii
Ctrl + 7	Przejdź do poprzedniej strony z błędem w strukturze partii
Ctrl + 8	Przejdź do następnej strony z błędem w strukturze partii
Ctrl + 9	Uruchom regułę sprawdzania poprawności bieżącej strony
Alt + L	Przejdź do następnego pola o niskiej pewności

Tabela 5. Skrót klawiszowy do funkcji drukowania

Skrót klawiszowy	Funkcja
P	Drukuj stronę
Shift+P	Drukuj dokument
Ctrl+P	Drukuj przedział stron

Temat nadrzędny: [Ułatwienia dostępu w programie Datacap Navigator](#)

Datacap

Datacap is a complete solution for document and data capture. Datacap scans, classifies, recognizes, validates, verifies, and exports data and document images quickly, accurately and cost effectively.

By combining the common recognition engines for OCR, ICR, OMR and barcodes with libraries of hundreds of script-based and code-based (.NET) actions, Datacap accurately captures data from any type of structured, highly variable, or unstructured documents.

Datacap can capture machine print, hand print, bar codes, and check box data. By using the Datacap rules engine, data capture can be tailored to fit the most demanding business requirements and can be changed quickly when business needs change.

For indexing applications, Datacap streamlines the manual data entry of index entries by using recognition to automatically identify the index values on each document and to automate the document identification process.

Datacap software components

Datacap contains various software components. Each component performs or supports a defined set of functions. When you run the installation program to install Datacap, you can select individual installation features. These features might contain one or more subfeatures.

List of Datacap software components – Custom installation

For custom installation, you can select following Datacap software components:

- Datacap Server
- Datacap Clients
 - Applications
 - IBM Datacap Accounts Payable
 - IBM Datacap Medical Claims
 - Datacap FastDoc
 - Datacap Studio
 - Datacap Maintenance Manager
- Datacap Rulerunner Server
- Datacap Web Server
- Datacap Web Service
- Datacap Windows Service
- Datacap Report Viewer
- Connectors
 - IBM Datacap Connector for eMail and Electronic Documents
 - IBM Datacap Connector for Fax
 - IBM Datacap Connector for EMC Documentum
 - IBM Datacap Connector for Microsoft SharePoint
- IBM Datacap Insight Edition (Document Analytics)

Note: The components with a prefix "IBM Datacap" are chargeable components, and you must buy them separately. In the installer, they appear with a red "X". You must select them and install, only if you are a valid user.

Datacap Server

The server component provides the following core functions of the Datacap system:

- Managing and serving batches to workstations and users
- Managing the tasks according to the workflow of the Datacap application
- Providing user authentication and access control, assigning batch IDs, controlling batch queues, and controlling access to the Datacap databases

All communication between the Datacap Server and its clients, or the other core server components, use the Datacap socket protocol. For communicating with the databases, Datacap Server uses Microsoft Object Linking and Embedding for Database (OLE DB). Datacap Server also uses the Common Internet File System (CIFS) interface to mount the file share that is required to access batches.

The Datacap Server supports various authentication systems. The main application settings file `datacap.xml` is on the file share where the application files are stored. In smaller systems, the file is on a Datacap Server. In medium or large systems, the file is on a dedicated file server or a NAS or SAN device.

Datacap Clients

The Datacap client component is a set of programs that you can use to access Datacap applications.

Applications

An application unites a set of Datacap capabilities with the aim of solving a specific business need. In Datacap custom installation setup, you can select following applications:

IBM Datacap Accounts Payable

The Datacap Accounts Payable solution is used for automating the process of capturing invoices. The APT uses optical character recognition and highly configurable location rules to accurately capture invoice data. The

invoice data comprises invoice number, invoice date, purchase order number, invoice total, and the data from each line item in the invoice. The invoice images and data are then delivered to your accounts payable, ERP, document management, and other systems. For more information, see [Datacap Accounts Payable](#).

IBM Datacap Medical Claims

Medical Claims is a software solution that automates data entry from professional claim forms (CMS 1500) that are used by individual medical providers or suppliers. The software also automates data from institutional claim forms (UB04) that are used by institutional providers such as hospitals. Medical Claims application is used to manage the entire capture process, including scanning of claims forms, enhancement of the images, recognition of data fields, and the validation and verification of data. The application exports images and index data to content management systems through integration interfaces. For more information, see [Medical Claims](#).

Datacap FastDoc

FastDoc is a client that you can use to scan, index, monitor jobs, and manually run background tasks on documents and image files. You can also use FastDoc as a rapid application development tool. You can run FastDoc in a stand-alone environment where FastDoc manages its own batches or as a client to the Datacap Server. For more information, see [FastDoc](#)

Datacap Studio

Datacap Studio is the application development environment for Datacap. When you install Datacap Studio, it includes an unrestricted version of the Application Manager, for managing multi-machine distributed environments through a centralized set of key Datacap configuration settings. These settings are stored in shared files. For more information, see [Datacap Studio](#).

Datacap Maintenance Manager

Datacap Maintenance Manager provides application monitoring and notification capabilities that can automate administrative functions such as resetting batches or archiving old batches. For more information, see [Installing and configuring Datacap Maintenance Manager](#).

Datacap Rulerunner Server

Datacap Rulerunner Service runs as a Windows service and runs batch processing tasks that do not require operator interaction, such as recognition and export. In a typical production environment, Rulerunner is configured to run the page identification and recognition tasks automatically. After verification and submission, Rulerunner detects that the batch is ready for export and runs the export task automatically. For more information, see [Installing Rulerunner on the Rulerunner server](#).

Datacap Web Server

Datacap Web Server hosts Datacap web applications with Microsoft IIS server. The IIS server communicates with the backend services through the Datacap Web Server. For more information, see [Configuring Datacap Web Server on a supported version of Windows Server](#).

Datacap Web Services

Datacap Web Services are also called wTM. It could be a windows-based service or Microsoft IIS-based web service. Datacap web service supports HTTP and HTTPS protocols. For more information, see [Installing Datacap Web Services](#).

Datacap Report Viewer

Datacap Report Viewer is a reporting tool for real-time reports of Datacap activity. Report Viewer retrieves usage statistics and other data from the Engine database. The Datacap Report Viewer web application displays real-time reports of activity that is related to your Datacap applications. For more information, see [Installing and configuring Datacap Report Viewer](#).

Connectors

Connectors are chargeable software components that you must buy separately. You must have a valid license to use the connectors.

IBM Datacap Connector for eMail and Electronic Documents

It imports email attachments from Exchange and Internet Message Access Protocol (IMAP) mail servers and converts electronic documents. For more information, see [Email Connector actions](#).

IBM Datacap Connector for Fax

It imports fax images from a Fax server. Fax Connector actions are used to create Datacap document batches from incoming faxes. You can also send the contents of a document to a specified fax number. For more information, see [Fax Connector actions](#).

IBM Datacap Connector for EMC Documentum

The Documentum Connector actions integrate Datacap applications with the Documentum Docbase content repository. You can then use the Documentum Connector actions to upload documents and index fields into a Documentum repository. For more information, see [Documentum Connector actions](#).

IBM Datacap Connector for Microsoft SharePoint

The Datacap Connector for Microsoft SharePoint actions integrates Datacap applications with Microsoft Office SharePoint Services for Microsoft SharePoint. You can use SharePoint Connector actions to upload documents and set index fields in a SharePoint library. For more information, see [SharePoint Connector actions](#)

IBM Datacap Insight Edition (Document Analytics)

Datacap Insight Edition is an intelligent capture solution that combines advanced imaging, natural language processing, and machine-learning technologies. It automates the processing of unstructured documents by performing multi-level analysis. Datacap Insight Edition can distinguish between different types of documents based on their structure. The solution is also capable of applying reasoning, logic, and context-sensitive analysis to identify and classify information.

Datacap Navigator and Datacap Web Client

Datacap Navigator and Datacap Web Client provide functions that are similar to the Datacap client but do not require more software to be installed on the computer.

When you verify a batch by using the web client, verification rules are run on the web server. You can also configure an application workflow and run administrative tasks such as setting up Datacap groups and users. Authentication for Datacap Navigator is done through calls to Datacap Web Services.

Datacap Navigator is based on IBM® Content Navigator technology and is installed, configured, and administered with IBM Content Navigator tools. Datacap Navigator communicates with the Datacap Server by using the Datacap Web Services APIs. For lookup and verification, Datacap Navigator uses the IBM Content Navigator External Data Services infrastructure.

Datacap databases

Datacap applications use relational databases. In the Datacap sample and add-on applications, Microsoft Access databases are used for portability reasons but must not be used in production. In a production system, Datacap databases are hosted in DB2®, Microsoft SQL Server, or Oracle. Datacap uses the following databases:

Administration

The Administration database stores information about users, groups, workstation, auditing, functional security, and application configuration. The administration database also stores workflow configurations.

Engine

The Engine database stores information about batches, statistics, and queue states.

Fingerprints

The Fingerprints database manages the pointers to the fingerprints that are used in an application. Each application has its own set of self-contained databases.

External

External database is a database that is accessed by Datacap applications during processing. Datacap applications can perform lookups to validate data such as vendor IDs, purchase order numbers, postal codes, customer IDs, on this database.

- [FastDoc](#)
FastDoc is a client that you can use to scan, index, monitor jobs, and manually run background tasks on documents and image files. You can also use FastDoc as a rapid application development tool. You can run FastDoc in a stand-alone environment where FastDoc manages its own batches or as a client to the Datacap Server.

FastDoc

FastDoc is a client that you can use to scan, index, monitor jobs, and manually run background tasks on documents and image files. You can also use FastDoc as a rapid application development tool. You can run FastDoc in a stand-alone environment where FastDoc manages its own batches or as a client to the Datacap Server.

By using FastDoc, you can do the following tasks:

- Automate the capture of index data from machine or manually printed documents and eliminate tedious and error-prone manual data entry.
- Capture index data from text and bar codes, retain document types and the variations in each document type and data field locations after it processes one time.
- Monitor batches in the Job Monitor where you can view batch details, change their status, and optionally delete batches.
- Separate pages into documents, automatically capture index entries for those documents, and export the index data and document content.
- Develop Datacap applications quickly by using your application as a starting point without using Datacap Studio or extend the application by using Datacap Studio if the FastDoc application does not support all your requirements.

You can create applications on FastDoc as a stand-alone client on your local computer, an integrated Datacap client, or both. At the Login screen, you can select Local to process a local workflow without connecting to

Datacap server. Or you can select Datacap to connect FastDoc to Datacap server and log into a Datacap application.

Local mode

In Local mode, FastDoc operates as an easy to use, stand-alone client that scans, manually indexes, and uploads batches of documents to repositories and file systems without connecting to Datacap server or other Datacap components. In Local mode, you can run smaller, less complicated jobs locally without setting up templates, programming rule sets, or configuring applications by using Datacap Studio and Application Manager on Datacap server.

Datacap mode

In Datacap mode, FastDoc scans, auto indexes, and uploads batches of documents to Datacap server. For example, you can use FastDoc to process invoice image files for the Datacap Accounts Payable application. In this mode, FastDoc can take advantage of Datacap server capabilities to further develop the application by using Datacap Studio and Application Manager. Click the user name at the top of the screen to logoff from Datacap and display the Log On screen. You can continue to work in Local mode or exit FastDoc.

Local and Datacap mode

You can run your batches locally and upload them to Datacap Server in the background later by using the Datacap Web Client Upload service.

FastDoc easily integrates with IBM® FileNet® Content Manager and Microsoft SharePoint.

In Local mode and Datacap mode, you can also export FastDoc documents to any of the repositories that are configured for Datacap in Datacap Studio. If you do not want to use these supported repositories, you can export documents as sets of images and index files in CSV or XML format to a file system in your network.

FastDoc works with document scanners and multifunction peripheral devices that use TWAIN and ISIS drivers. It can also process previously scanned and faxed images and multi-page files in TIFF, JPEG, BMP, PNG, and PDF formats.

You can run FastDoc Administrator mode and Operator mode. In Administrator mode, you can configure document and processing settings and use the batch profile for RAD. Operator mode is used for scanning and processing documents in production.

Parent topic: [Datacap software components](#)

Related tasks:

[Installing and configuring the Datacap Web Client upload service](#)

Datacap capture solutions

Datacap provides several industry-specific capture applications that are available out of the box.

- [Medical Claims](#)
Medical Claims is a software solution that automates data entry from professional claim forms (CMS 1500) used by individual medical providers or suppliers, and from institutional claim forms (UB04) used by institutional providers such as hospitals.
- [Datacap Accounts Payable](#)
The Datacap APT foundation application is the Datacap Accounts Payable solution for automating the process of capturing invoices.

Medical Claims

Medical Claims is a software solution that automates data entry from professional claim forms (CMS 1500) used by individual medical providers or suppliers, and from institutional claim forms (UB04) used by

institutional providers such as hospitals.

Medical Claims manages the entire capture process, from scanning of claims forms, enhancement of the images to increase recognition accuracy, recognition of data fields, and the validation and verification of data. Medical Claims then coordinates the upload of HIPAA-compliant claim data to adjudication systems for payment, and exports images and index data to content management systems through integration interfaces. It eliminates costly, error-prone manual data entry and accelerates claim processing.

Datacap Studio provides an enhanced environment for creating and maintaining rules. Claim forms can be captured by using a scanner or other capture device, or can be previously scanned for the digital images to be saved to a folder. Professional (CMS 1500) forms and institutional (UB04) forms are now processed in separate workflows, and in these workflows, red dropout (data alone) claims and black (with lines and boxes) claims must be batched and scanned separately.

Medical Claims recognizes machine print data (OCR), handprint data (ICR), and bar codes on the forms. Extensive validations are then applied to the claim data to ensure that accuracy. Validations include lookups for fields such as state codes, member IDs, diagnostic codes, and place of service. Other validations check for required fields and appropriate data formats. The form data and snippets of the claim image are then presented in a verification panel for an operator to review and, if necessary, make corrections. Fields that need attention are automatically highlighted for the operator. Verification panels can be in Windows based or browser-based. After the operator reviews the form, the data verification is again applied to ensure that accuracy. Medical Claims then creates EDI 837 formatted data files from the claims data. The formats of these files are typically customized for the target business and adjudication systems.

Parent topic: [Datacap capture solutions](#)

Datacap Accounts Payable

The Datacap APT foundation application is the Datacap Accounts Payable solution for automating the process of capturing invoices.

APT uses optical character recognition and highly configurable location rules to accurately capture invoice data such as invoice number, invoice date, purchase order number, invoice total, and the data from each line item in the invoice. The invoice images and data are then delivered to your accounts payable, ERP, document management, and other systems.

Datacap Accounts Payable (Datacap APT) is delivered with sample images and a multi-page invoice separator page for different preconfigured jobs. You can run the tasks in these jobs to use APT without having to configure additional steps.

Datacap APT includes the APT Add Demo Vendor utility program for you to add vendors to a vendor list, which acts as a sample Accounts Payable Vendor database. The vendor list is used to perform vendor lookups and validations on your Accounts Payable Vendor database because APT cannot initially access the database.

Parent topic: [Datacap capture solutions](#)

Locating applications on a network

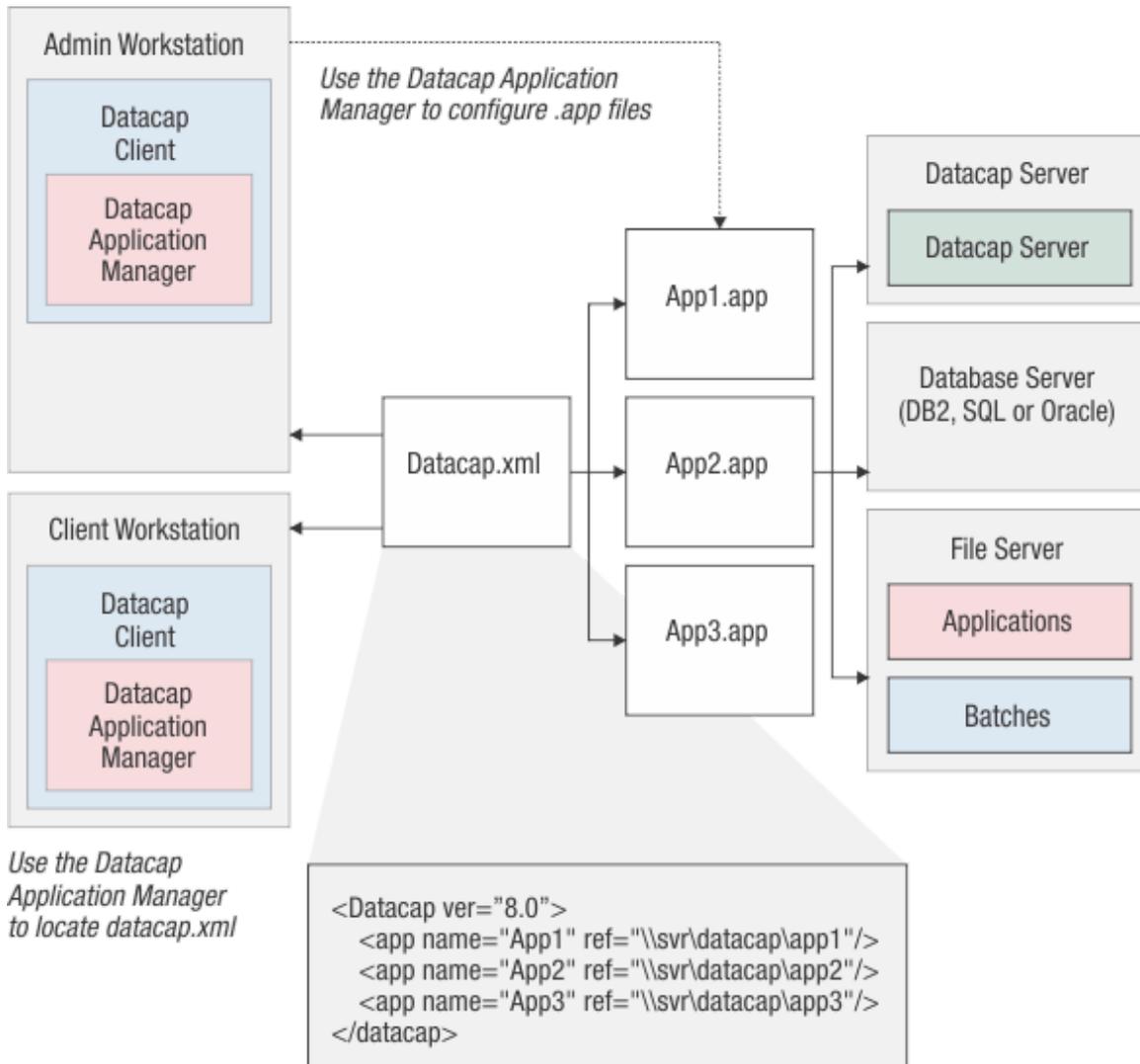
The Datacap datacap.xml file, which is typically in the shared Datacap folder on the Datacap server, contains information about the locations of the available Datacap applications.

About this task

Each Datacap application has its own application configuration file that identifies the locations of the components that are used by that application. This application configuration file (.app) includes details about the locations of its components, such as batches folder, input image folder, and databases.

The Datacap Application Manager is used to identify the location of the datacap.xml file and to manage the contents of the application configuration files.

In the diagram, the configuration file for App2 defines the location of the Datacap server that hosts the application and the locations of the application's databases, batches folder, and other components of the application.



Scenario: An imaging solution to streamline document storage

Datacap integrates with IBM® FileNet® Content Manager and IBM Case Foundation to provide a comprehensive imaging solution.

Problem

Efficiently managing the documents that they need to conduct their businesses is a challenge that many organizations face today. Storing and archiving paper and electronic documents is a problem of immense proportions. The challenge is not only to control all types of media, but also to ensure that you can retrieve and

manage the data throughout the lifecycle of business processes. A production imaging solution provides an optimal way of addressing these challenges.

Solution

Finding a way to overcome the problems that are inherent in paper and electronic storage aligns with organizational efforts to use enterprise content management systems. Those systems that can include production imaging features are compelling.

For a complete discussion of imaging solution scenarios, see *Implementing Imaging Solutions with IBM Production Imaging Edition and IBM Datacap* at <http://www.redbooks.ibm.com/abstracts/sg247969.html?Open>.

Parent topic: [Planning your Datacap system](#)

Scenario: Order processing automation for a mail-order pharmacy

Datacap can integrate with IBM® FileNet® Content Manager to provide comprehensive pharmacy order processing by mail.

Problem

Many mail-order businesses, such as pharmacies, require improved image quality on incoming forms, more automated processes, increased performance and throughput, and an overall reduction of manual labor.

Solution

In the initial steps of a solution, Datacap provides an excellent way to scan, recognize, and verify incoming documentation related to new and existing orders. Various connectors can be licensed with the product to enable storage integration with document and image storage products, such as IBM FileNet Content Manager.

Specifically, an organization can deploy Datacap software to capture information from mail-order forms by using optical character recognition (OCR), intelligent character recognition (ICR), optical mark recognition (OMR), and barcode recognition. The captured information is used to automate the indexing of document images for storage to a Content Manager system. The data is also used to populate an additional database for creating prescription fulfillment at the production center. There, workers fill and package the captured prescription orders which are then mailed to customers around the nation.

For more details on an actual customer success story related to automating mail-order pharmacy order processing with an IBM based solution, see http://www.ibm.com/software/success/cssdb.nsf/CS/JHAL-8BPQQP?OpenDocument&Site=cmportfolio&cty=en_us. For details on how IBM Case Manager might be used to extend such a solution to provide a full-scale healthcare-related case management system, see https://www.ibm.com/developerworks/mydeveloperworks/blogs/iic-san-mateo/entry/test_entry?lang=en.

Parent topic: [Planning your Datacap system](#)

Uwagi

Niniejsza publikacja została opracowana z myślą o produktach i usługach oferowanych w Stanach Zjednoczonych.

IBM może nie oferować w innych krajach produktów, usług lub opcji omawianych w tej publikacji. Informacje o produktach i usługach dostępnych w danym kraju można uzyskać od lokalnego przedstawiciela IBM. Odwołanie do produktu, programu lub usługi IBM nie oznacza, że można użyć wyłącznie tego produktu, programu lub usługi. Zamiast nich można zastosować ich odpowiednik funkcjonalny pod warunkiem, że nie narusza to praw własności intelektualnej IBM. Jednakże cała odpowiedzialność za ocenę przydatności i sprawdzenie działania produktu, programu lub usługi pochodzących od producenta innego niż IBM spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Przedstawienie niniejszej publikacji nie daje żadnych uprawnień licencyjnych do tychże patentów. Pisemne zapytania w sprawie licencji można przesyłać na adres:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Zapytania w sprawie licencji na informacje dotyczące zestawów znaków dwubajtowych (DBCS) należy kierować do lokalnych działów własności intelektualnej IBM (IBM Intellectual Property Department) lub zgłaszać na piśmie pod adresem:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japonia

Poniższy akapit nie obowiązuje w Wielkiej Brytanii, a także w innych krajach, w których jego treść pozostaje w sprzeczności z przepisami prawa miejscowego: INTERNATIONAL BUSINESS MACHINES CORPORATION DOSTARCZA TĘ PUBLIKACJĘ W TAKIM STANIE, W JAKIM SIĘ ZNAJDUJE ("AS IS"), BEZ JAKICHKOLWIEK GWARANCJI (W TYM TAKŻE RĘKOJMI), WYRAŻNYCH LUB DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU ORAZ GWARANCJI, ŻE PUBLIKACJA TA NIE NARUSZA PRAW STRON TRZECICH. Ustawodawstwa niektórych krajów nie dopuszczają zastrzeżeń dotyczących gwarancji wyraźnych lub domniemanych w odniesieniu do pewnych transakcji; w takiej sytuacji powyższe zdanie nie ma zastosowania.

Informacje zawarte w niniejszej publikacji mogą zawierać nieścisłości techniczne lub błędy typograficzne. Informacje te są okresowo aktualizowane, a zmiany te zostaną uwzględnione w kolejnych wydaniach tej publikacji. IBM zastrzega sobie prawo do wprowadzania ulepszeń i/lub zmian w produktach i/lub programach opisanych w tej publikacji w dowolnym czasie, bez wcześniejszego powiadomienia.

Wszelkie wzmianki w tej publikacji na temat stron internetowych innych podmiotów zostały wprowadzone wyłącznie dla wygody użytkownika i w żadnym wypadku nie stanowią zachęty do ich odwiedzania. Materiały dostępne na tych stronach nie są częścią materiałów opracowanych dla tego produktu IBM, a użytkownik korzysta z nich na własną odpowiedzialność.

IBM ma prawo do używania i rozpowszechniania informacji przysłanych przez użytkownika w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Licencjobiorcy tego programu, którzy chcieliby uzyskać informacje na temat programu w celu: (i) wdrożenia wymiany informacji między niezależnie utworzonymi programami i innymi programami (łącznie z tym opisywanym) oraz (ii) wspólnego wykorzystywania wymienianych informacji, powinni skontaktować się z:

IBM Corporation
J46A/G4

555 Bailey Avenue
San Jose, CA 95141-1003
USA

Informacje takie mogą być udostępnione, o ile spełnione zostaną odpowiednie warunki, w tym, w niektórych przypadkach, zostanie uiszczona stosowna opłata.

Licencjonowany program opisany w niniejszej publikacji oraz wszystkie inne licencjonowane materiały dostępne dla tego programu są dostarczane przez IBM na warunkach określonych w Umowie IBM z Klientem, Międzynarodowej Umowie Licencyjnej IBM na Program lub w innych podobnych umowach zawartych między IBM a użytkownikami.

Wszelkie dane dotyczące wydajności zostały zebrane w kontrolowanym środowisku. W związku z tym rezultaty uzyskane w innych środowiskach operacyjnych mogą się znacząco różnić. Niektóre pomiary mogły być dokonywane na systemach będących w fazie rozwoju i nie ma gwarancji, że pomiary wykonane na ogólnie dostępnych systemach dadzą takie same wyniki. Niektóre z pomiarów mogły być estymowane przez ekstrapolację. Rzeczywiste wyniki mogą być inne. Użytkownicy powinni we własnym zakresie sprawdzić odpowiednie dane dla ich środowiska.

Informacje dotyczące produktów innych niż produkty IBM pochodzą od dostawców tych produktów, z opublikowanych przez nich zapowiedzi lub innych powszechnie dostępnych źródeł. IBM nie testował tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące możliwości produktów innych podmiotów należy kierować do dostawców tych produktów.

Wszelkie stwierdzenia dotyczące przyszłych kierunków rozwoju i zamierzeń IBM mogą zostać zmienione lub wycofane bez powiadomienia.

Publikacja ta zawiera przykładowe dane i raporty używane w codziennej pracy. W celu kompleksowego ich zilustrowania podane przykłady zawierają nazwiska osób prywatnych, nazwy przedsiębiorstw oraz nazwy produktów. Wszystkie te nazwy/nazwiska są fikcyjne i jakiegokolwiek podobieństwo do istniejących nazw/nazwisk i adresów jest całkowicie przypadkowe.

LICENCJA W ZAKRESIE PRAW AUTORSKICH:

Niniejsza publikacja zawiera przykładowe aplikacje w kodzie źródłowym, ilustrujące techniki programowania w różnych systemach operacyjnych. Użytkownik może kopiować, modyfikować i dystrybuować te programy przykładowe w dowolnej formie bez uiszczania opłat na rzecz IBM, w celu projektowania, używania, sprzedaży lub dystrybucji aplikacji zgodnych z aplikacyjnym interfejsem programowym dla tego systemu operacyjnego, dla którego napisane zostały programy przykładowe. Programy przykładowe nie zostały gruntownie przetestowane. IBM nie może zatem gwarantować ani sugerować niezawodności, użyteczności i funkcjonalności tych programów. Programy przykładowe są dostarczane w stanie, w jakim się znajdują ("AS IS"), bez udzielania jakichkolwiek gwarancji (rękojmię również wyłącza się). IBM nie ponosi odpowiedzialności za jakiegokolwiek szkody wynikające z używania programów przykładowych.

Każda kopia programu przykładowego lub jakiegokolwiek jego fragment, jak też jakiegokolwiek prace pochodne muszą zawierać następujące uwagi dotyczące praw autorskich: © (nazwa przedsiębiorstwa użytkownika, rok). Fragmenty niniejszego kodu pochodzą z programów przykładowych IBM Corp. © Copyright IBM Corp. 2004, 2010. Wszelkie prawa zastrzeżone.

W przypadku przeglądania niniejszych informacji w formie elektronicznej zdjęcia i kolorowe ilustracje mogą nie być wyświetlane.

- [Postanowienia dotyczące ochrony prywatności](#)
- [Znaki towarowe](#)

Postanowienia dotyczące ochrony prywatności

Oprogramowanie IBM, w tym rozwiązanie SaaS (Software as a Service), zwane dalej "Oferowanym Oprogramowaniem" może korzystać z informacji cookie lub z innych technologii do gromadzenia danych o używaniu produktów, do poprawienia jakości usług dla użytkowników końcowych, do dopasowania interakcji do ich oczekiwań oraz do innych celów. W wielu przypadkach Oferowane Oprogramowanie nie gromadzi informacji pozwalających na identyfikację osoby. Część Oferowanego Oprogramowania może jednak umożliwiać gromadzenie informacji pozwalających na identyfikację osoby. Jeśli Oferowane Oprogramowanie korzysta z informacji cookie do gromadzenia informacji pozwalających na identyfikację osoby, poniżej znajdują się szczegółowe informacje na temat takiego korzystania.

Oferowane Oprogramowanie nie korzysta z informacji cookie ani innych technologii do gromadzenia informacji pozwalających na identyfikację osoby.

Jeśli konfiguracje Oferowanego Oprogramowania umożliwiają gromadzenie informacji pozwalających na identyfikację użytkowników końcowych za pośrednictwem informacji cookie lub innych technologii, należy wystąpić o poradę prawną w zakresie prawa obowiązującego przy takim gromadzeniu danych, w tym wymagań dotyczących powiadomienia i zgody.

Więcej informacji na temat korzystania z różnych technologii, w tym z informacji cookie, do opisanych wyżej celów znajduje się w sekcji Ochrona prywatności w IBM, po adresem <http://www.ibm.com/privacy> oraz Oświadczenie IBM o Ochronie Prywatności w Internecie, pod adresem <http://www.ibm.com/privacy/details>, a także w sekcji zatytułowanej "Cookies, Web Beacons and Other Technologies" oraz "IBM Software Products and Software-as-a-Service Privacy Statement", pod adresem <http://www.ibm.com/software/info/product-privacy>.

Temat nadrzędny: Uwagi

Znaki towarowe

Następujące nazwy są znakami towarowymi International Business Machines Corporation w Stanach Zjednoczonych i/lub innych krajach: <http://www.ibm.com/legal/copytrade.shtml>

Adobe, logo Adobe, PostScript i logo PostScript są zastrzeżonymi znakami towarowymi lub znakami towarowymi Adobe Systems Incorporated w Stanach Zjednoczonych i/lub w innych krajach.

Intel, logo Intel, Intel Inside, logo Intel Inside, Intel Centrino, logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium i Pentium są znakami towarowymi lub zastrzeżonymi znakami towarowymi Intel Corporation lub przedsiębiorstw podporządkowanych w Stanach Zjednoczonych i innych krajach.

Linux jest zastrzeżonym znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i w innych krajach.

Microsoft, Windows i Windows NT są znakami towarowymi Microsoft Corporation w Stanach Zjednoczonych i/lub w innych krajach.

UNIX jest zastrzeżonym znakiem towarowym The Open Group w Stanach Zjednoczonych i innych krajach.

Java™ oraz wszystkie znaki towarowe i logo dotyczące języka Java są znakami towarowymi lub zastrzeżonymi znakami towarowymi Oracle i/lub przedsiębiorstw afiliowanych.

iOS jest zastrzeżonym znakiem towarowym Cisco w Stanach Zjednoczonych używanym przez Apple Computer Corp w ramach licencji.

Android jest znakiem towarowym Google Inc.

Prawa autorskie do ikon z rodziny Material należą do Google Inc. Ikony pochodzą ze strony <https://www.google.com/design/icons/>

Nazwy innych firm, produktów i usług mogą być znakami towarowymi lub znakami usług innych podmiotów.

Temat nadrzędny: [Uwagi](#)

Installing

Before you install or migrate your system, install the necessary prerequisites and plan a system that meets your needs.

- [Planning your Datacap system](#)
Planning your Datacap system includes activities that are related to domains, Windows accounts, authentication systems, and Datacap users, groups, stations, and databases.
- [Prerequisites for installing Datacap](#)
Before you begin any installation of Datacap, ensure that your system is configured with the required prerequisite software.
- [Installing and configuring Datacap on one machine](#)
You can install, configure, and operate Datacap on a single machine for demonstration or testing purposes only.
- [Installing and configuring in a client/server environment](#)
The typical installation configuration for Datacap is one where the various Datacap software components are installed on dedicated machines. These software components include application servers, web servers, database servers, scan workstations, verify workstations, etc.
- [Configuring databases](#)
The Datacap applications can use DB2®, Microsoft SQL Server, or Oracle databases to store workflow definitions, security parameters, processing information, and fingerprints.
- [Upgrading](#)
Migrating a Datacap environment to a new release requires that you upgrade the component software and migrate your deployed applications. Depending on the currently installed version of Datacap, upgrading the software may first require removing the previous version before installing the new release. The migration steps can include revising existing functionality, adding new features, or changing baseline user interfaces.
- [Migrating from previous releases](#)
Migrating a Datacap environment to a new release requires that you upgrade the component software and migrate your deployed applications. The migration steps can include revising existing functionality, adding new features, or changing baseline user interfaces.
- [Uninstalling Datacap](#)
The Datacap installation removal process removes only those files that the previous Datacap installation process created.

Planning your Datacap system

Planning your Datacap system includes activities that are related to domains, Windows accounts, authentication systems, and Datacap users, groups, stations, and databases.

About this task

Review these topics so that you understand the following concepts before you install and configure IBM® Datacap:

Datacap domain and Windows accounts

In a client/server environment, Datacap users, background services, and processes require read, write, create, delete, and change files on various computers. To grant the appropriate permissions to the appropriate users, you must create Windows accounts for the Datacap users, application pools, background services, and background processes.

In addition, the computers on which Datacap is installed and to which the Windows accounts are granted access rights must be part of a single domain or a set of trusted domains.

A number of the Windows accounts used by Datacap background services must also be granted the Logon as a service on the domain controller. For more information, see [Domains and Windows accounts](#).

Authentication systems

You also need to determine how you want Datacap users to be authenticated. Datacap supports the use of the following internal and external authentication systems:

- Datacap authentication (TMA)
- Active Directory Service Interfaces (ADSI)
- Lightweight Directory Access Protocol (LDAP)
- Active Directory Lightweight Directory Services (ADLDS)
- Low-Level Lightweight Directory Access Protocol (LLLDAP)

While Datacap can access servers that host ADLDS or LLLDAP authentication systems that are outside the Datacap domain, ADSI and LDAP authentication servers must be in the Datacap domain. For more information, see [Authentication](#).

Datacap users, groups, and stations

For Datacap to use its authentication system, you must set up users and stations in your Datacap application. If you choose to manage permissions at the group level, you can also set up groups. For Datacap to use an external ADSI, LDAP, ADLDS, or LLLDAP authentication system, you must set up users or groups in your Datacap application that correspond to the Datacap accounts or groups that are set up in the external authentication system. For more information, see [Users and groups](#).

Stations in a Datacap application can be set up as you choose. You can set up a station for each physical computer, or a single station name can be used for multiple computers. You can set up stations to limit the types of tasks that can be run by a user and provide routing of batches. You can also monitor individual or aggregated activity. For more information, see [Stations](#).

Datacap databases

Because Datacap applications include the use of databases, the Windows accounts of some Datacap components must be set up as database users and assigned the appropriate database permissions. For more information, see [Database users](#).

- [System requirements](#)
Use the Software Product Compatibility Reports page to generate a report for supported operating systems, related software, hypervisors, hardware requirements, and detailed system requirements, including component-level details. You can also get support information related to product translations and product end of service.
- [Planning your system architecture](#)
Datacap provides a flexible and scalable architecture for distributing tasks across machines according to the anticipated processing load.
- [Domains and Windows accounts](#)
In a client/server environment, regardless of the authentication system in use, all of the computers that are hosting the Datacap software components must be part of a single domain or part of multiple

domains that trust each other. You need to know the name of your domain or domains during some steps of the Datacap installation process.

- [Stations](#)
Station names in a Datacap application can be set up for each physical computer, for each type of workstation (such as Scan or Verify), or, for a group of computers in a department or a physical location.
- [Databases](#)
Datacap is installed with a Microsoft Access database, however, Microsoft Access is not supported in a Datacap production environment. You can configure your Datacap production environment to use a DB2®, Microsoft SQL Server, or Oracle database.
- [Installation methods](#)
The installation plan for Datacap depends on whether you are upgrading from an earlier version, and whether you are installing to the default location on the machine, C:\Datacap.
- [High availability \(load balancing\)](#)
You can use network load balancers to manage client requests across servers in a Datacap system.
- [Scenario: An imaging solution to streamline document storage](#)
Datacap integrates with IBM FileNet® Content Manager and IBM Case Foundation to provide a comprehensive imaging solution.
- [Scenario: Order processing automation for a mail-order pharmacy](#)
Datacap can integrate with IBM FileNet Content Manager to provide comprehensive pharmacy order processing by mail.

Parent topic: [Installing](#)

System requirements

Use the Software Product Compatibility Reports page to generate a report for supported operating systems, related software, hypervisors, hardware requirements, and detailed system requirements, including component-level details. You can also get support information related to product translations and product end of service.

Go to the page at [Software Product Compatibility Reports](#) to create a high-level report for operating systems, related software, hypervisors, and supported translations for Datacap. You can also create an in-depth report to get detailed Datacap system requirements, hardware requirements, and end of service information. You can search for Datacap in all of the report types and reports are generated based on your query values.

The following report types are the most commonly generated reports from software product compatibility reports:

Detailed system requirements

When you select your product version for the detailed system requirements report, you can set a report filter for Operating system platforms, Product components, and Capabilities, including prerequisites and support software. After you view the report, you can save it as a URL to generate anytime or download it as a PDF.

Hardware requirements

When you select your product version for the hardware requirements report, you can set a report filter by the Operating system families option. Set the operating system filter by selecting some or all of the operating systems that are supported by your product. After you view the report, you can save it as a URL to generate anytime or download it as a PDF.

Translations

You can search for the list of available translations by a specific product or for a list of products that are translated into a specific language. For example, when you select Translations available for a specific product and enter the product and version, you can see a list of the available translations for that product version. After you view the report, you can save it as a URL to generate anytime or download it as a PDF.

End of service

The end of service report shows the service window of the products that you specify over an eight-year span. For example, you can find out when your product is scheduled to go out of service.

Parent topic: [Planning your Datacap system](#)

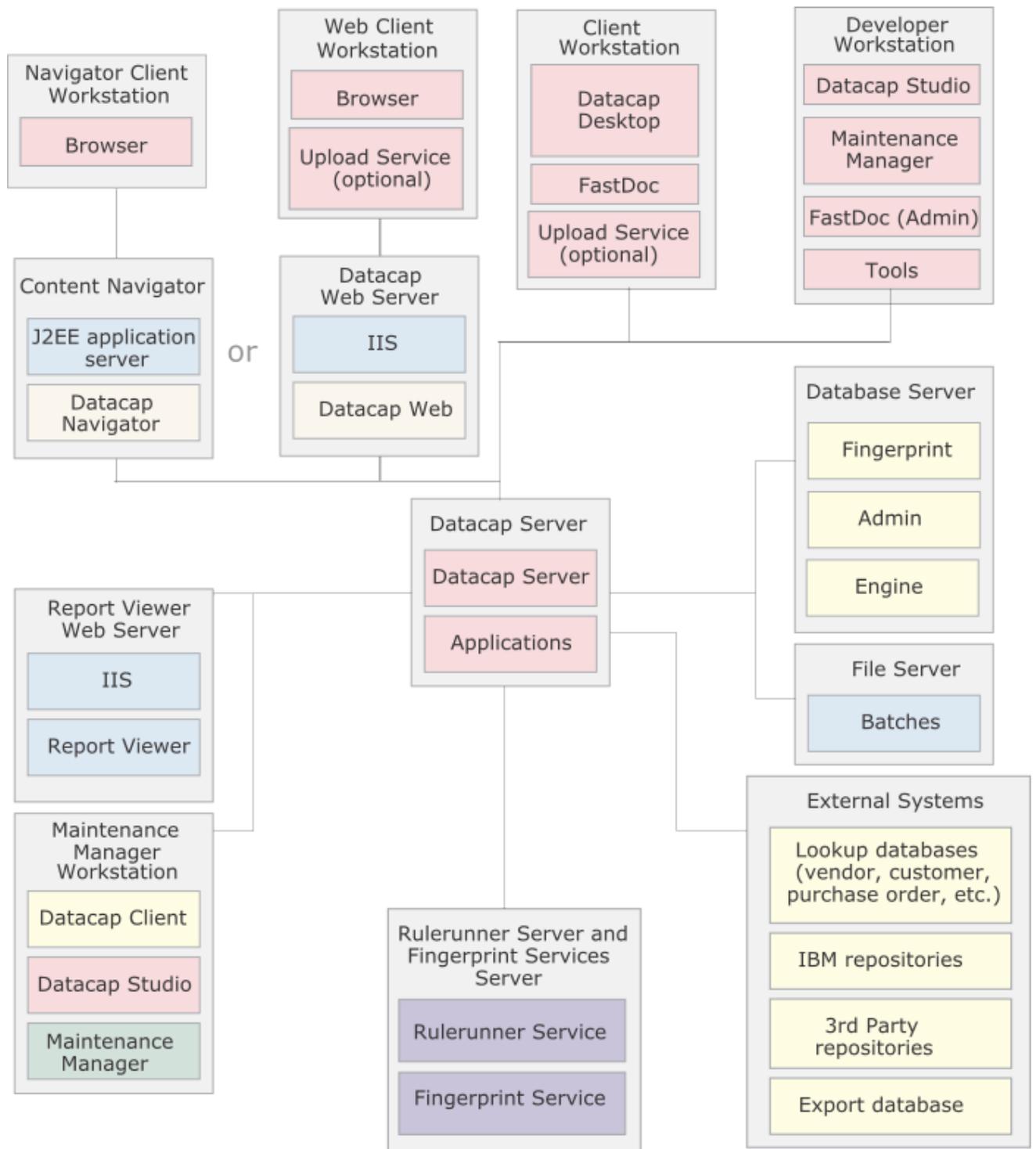
Planning your system architecture

Datacap provides a flexible and scalable architecture for distributing tasks across machines according to the anticipated processing load.

About this task

At one end of the spectrum is the single machine configuration, where all Datacap software components are installed on the same machine. This configuration is typically used for providing product demonstrations, in a proof of concept environment, or during initial product evaluation.

At the other end of the spectrum is the client/server configuration, where the various Datacap software components are installed on dedicated machines, such as web servers and database servers. This configuration can support hundreds of simultaneous users, and uses centralized application management and shared databases.



Spanning the center of the spectrum are various hybrid configurations in which two or more Datacap software components are installed on the same machine. For example, you might install and run Datacap Web Client and Report Viewer on the same web server. You might also install and run the Rulerunner Service and the Fingerprint Service on another server.

Restriction: Datacap applications can be configured for use with Datacap Web Client or Datacap Navigator, but not both.

- [Application Manager](#)

When you create a new application by using the application wizard, the application is added to the list of applications in the Application Manager. The Application Manager is used to identify the location of the datacap.xml file and to manage the contents of the application configuration file. The application

configuration file includes details about the locations of its components, such as the batches folder, the input image folder, and databases.

- [Rulerunner Manager](#)
On the Rulerunner Manager, you can configure settings for how the Rulerunner Service processes tasks. You can decide which applications, workflows, and tasks are processed by the Rulerunner Service, how a batch is selected for processing, and configure Rulerunner logging.
- [Datacap Server Manager](#)
In the Datacap Server Manager, you can start, pause, and stop the Datacap Server service, set the System event log and the Datacap log, and determine whether to set the queue by job or task.

Parent topic: [Planning your Datacap system](#)

Application Manager

When you create a new application by using the application wizard, the application is added to the list of applications in the Application Manager. The Application Manager is used to identify the location of the `datacap.xml` file and to manage the contents of the application configuration file. The application configuration file includes details about the locations of its components, such as the batches folder, the input image folder, and databases.

You can use the Application Manager to store passwords, connection strings, paths, and other settings in the `.app` application configuration file. Each Datacap application has its own application configuration file that identifies the locations of the components that are used by that application. You must use the Application Manager to modify the application configuration file. Do not modify the `.app` file directly. For example, you can copy an application from a test environment to a production environment, except for the `.app` file that is unique for each environment.

The Application Manager works well with separate test and production environments because it stores information that is specific to the application environment, such as database locations and physical paths. You must use the Application Manager to configure the export database.

Application settings

On the Main tab, you must enter the application paths for the batch, export, and Fingerprint folders. You must also enter the workflow paths, such as the Setup DCO, Locale, Rules, and various database paths.

On the Datacap tab, you must enter the paths for the Administration and Engine databases and the server name and location.

On the Rulerunner tab, you can enter tasks and task profiles to be run by Rulerunner. Rulerunner runs the task profiles that are listed in the Application Manager.

On the Custom values tab, developers can set up custom values to create application-specific settings and your application can access those settings by using smart parameters. You can also store passwords in the `.app` application file. To pass passwords as action parameters, you can use smart parameters that retrieve credentials from the `.app` file where the passwords are stored as encoded strings. Passwords must be entered into the Advanced Values section to be encrypted.

On the Service tab, you must enter the path to the `datacap.xml` application management file.

Parent topic: [Planning your system architecture](#)

Related information:

[Storing passwords, connection strings, and other parameters in the `.app` file](#)

[Configuring the export database](#)

[Smart parameters](#)

Rulerunner Manager

On the Rulerunner Manager, you can configure settings for how the Rulerunner Service processes tasks. You can decide which applications, workflows, and tasks are processed by the Rulerunner Service, how a batch is selected for processing, and configure Rulerunner logging.

Rulerunner Manager settings

On the Rulerunner tab, you can start and stop the Rulerunner Service. When you click Stop, the Rulerunner Service stops after the current processing is complete.

On the Rulerunner Login tab, you must select the authentication method when you want to change the Rulerunner Service settings and when the Rulerunner Service runs tasks.

On the Workflow:Job:Task tab, you can configure the applications and tasks that the Rulerunner Service processes. In the workflow pane, you can see the list of applications that are stored in the datacap.xml file on the Datacap server and information about the application workflows, jobs, and tasks. In the thread pane, you can create threads and configure the tasks that each Rulerunner thread is to process. You can create one or more threads and copy application tasks to the threads. The details pane displays the settings that are related to the node that is selected in the first two panes. You can specify the priority of the task and set a value to delay a task that is not available for processing.

On the Settings tab, you can set the Thread Timeout with the maximum time the Rulerunner Service waits until stopping all Rulerunner threads. You can also set the time interval for restarting the Rulerunner service. The Mixed Queuing option determines how a batch is selected to run for an application. If you clear the Mixed Queuing option, the queuing is sequential.

- Mixed queuing - The Rulerunner service sends the list of tasks to run within the application to the Datacap server. The Datacap server selects the oldest, highest priority batch that is pending for the selected tasks and notifies Rulerunner to run that batch.
- Sequential queuing - The Rulerunner service selects the oldest pending batch from each of the selected tasks, one after the other. The first batch is selected from the first task. The second batch is selected from the second task. Sequential queuing is typically required when the Rulerunner service runs batch creation tasks, such as Vscan.

On the Logging tab, there are four options to set up Rulerunner logging. You can select the Quick Log, ATM Log, Rulerunner Log, or RRS Log option to set the level and detail that you want in the log.

- Select the Quick Log when you want to adjust the level of detail that is written to all of the logs: the ATM log, the Rulerunner log, and the RRS log. When you change the logging level on the Quick Log, the logging level is adjusted for all three logs.
- Select the ATM Log when you want to adjust the type and amount of detail that is written only to the ATM log, including the time difference between the current and the last message.
- Select the Rulerunner Log when you want to adjust the type and amount of detail that is written to the Rulerunner and thread-specific log files and to the System Event log.
- Select the RRS Log when you want to adjust the type and amount of detail that is written to the RRS log, including the message severity level, date and time stamp, and the application ID.

If the Rulerunner Service loses its connection to the Datacap server, the service logs exception messages that indicate the lost connection. The frequency of these logged messages depends on the interval between reconnection attempts. The default interval is 300 seconds. You can change this interval by setting the value of

the Windows registry key Sleep reconnect timeout. The registry location for the key is HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Datacap\Rulerunner\Misc\.

Parent topic: [Planning your system architecture](#)

Related tasks:

[Configuring Rulerunner to run tasks](#)

Datacap Server Manager

In the Datacap Server Manager, you can start, pause, and stop the Datacap Server service, set the System event log and the Datacap log, and determine whether to set the queue by job or task.

In the Advanced settings pane of the Datacap Server Manager, you can configure the Datacap Server service connections port and set the database command timeout. You can also select your authentication method and configure the Batch naming template. For more information, see *Advanced settings for the Datacap Server service*.

Parent topic: [Planning your system architecture](#)

Related reference:

[Datacap Server service settings](#)

[Advanced settings for the Datacap Server service](#)

Domains and Windows accounts

In a client/server environment, regardless of the authentication system in use, all of the computers that are hosting the Datacap software components must be part of a single domain or part of multiple domains that trust each other. You need to know the name of your domain or domains during some steps of the Datacap installation process.

Additionally, you must create Windows accounts for the Datacap users, application pools, and background services and processes that read, write, create, delete, and change files and folders on various computers.

Datacap does not require you to create individual Windows accounts for every user and background service or process that you identify. You can choose to use a single Windows account for a group of users who all need the same sharing and security permissions. For example, all instances of the Rulerunner Service can use the same Windows account.

The permissions that you grant to each type of Windows account varies. Specific instructions for granting permissions to each Windows account for each type of user, background service, or background process are provided in the detailed installation and configuration instructions for that user, background service, or background process.

In general, the following Datacap software components are associated with types of users, background services and processes, and application pools.

Users that use interactive Datacap software components

Datacap Desktop, FastDoc, Datacap Studio, Application Manager, Datacap Web Client, and others.

Background services and processes

Rulerunner Service, Datacap Web Client Upload Service, Datacap Web Services, Maintenance Manager, Datacap Server Service.

Application pools

Datacap Web Client, Report Viewer, Fingerprint Service.

- [Users and groups](#)
You can define the tasks that can be run by specific users or groups of users. You can also define user names to control the flow of batches to specific users.
- [Database users](#)
The Datacap applications access the Administration and Engine databases and can optionally include Fingerprint databases. You must grant access rights to the Datacap components for the databases that they use.
- [Authentication](#)
You must determine which authentication system you are using and configure Datacap for user authentication or group authentication.

Parent topic: [Planning your Datacap system](#)

Users and groups

You can define the tasks that can be run by specific users or groups of users. You can also define user names to control the flow of batches to specific users.

For each of your Datacap applications, identify the Datacap software components that you are using and the associated users. Be sure to include both human users and automatic processes.

Include human users that use interactive Datacap software components, such as Datacap Desktop, FastDoc, Datacap Web Client, Datacap Server Manager, Application Manager, and Rulerunner Manager. These users are typically system administrators, developers, supervisors, and other types of users.

Include all background services and processes that run Datacap tasks automatically, such as Rulerunner, Datacap Web Client Upload Service, Datacap Web Services, and Datacap Maintenance Manager.

Parent topic: [Domains and Windows accounts](#)

Database users

The Datacap applications access the Administration and Engine databases and can optionally include Fingerprint databases. You must grant access rights to the Datacap components for the databases that they use.

Datacap applications can also include access to external databases that are used for lookups (Lookup) and to which data is exported (Export).

When all of these databases (Administration, Engine, Fingerprint, Lookup, Export) are used by a Datacap application, each Datacap component must be granted specific access rights to the appropriate databases.

Table 1. List of Datacap components and the databases they access

Datacap component	Access these databases
Datacap Server	Administration, Engine, Fingerprint, Lookup
Rulerunner	Fingerprint, Lookup, Export
Datacap Web Client, Datacap Desktop	Fingerprint and Lookup - when you run actions, such as validation task profiles, that use these databases
FastDoc	Lookup - when you run actions, such as validation task profiles, that use this database

When Microsoft SQL Server, Oracle, or DB2® authentication is used for the database connection strings, then a single database account can be used for connections to each database. Create these credentials for this account (user ID and password) in the Datacap Application Manager.

When Windows authentication is used with SQL Server, then there must be one SQL Server account for each Windows account that is used by a Datacap component. Alternatively, Windows groups can be used to authenticate the Datacap component Windows account to SQL Server.

When you are using an authentication system that does not require users to be set up in your Datacap application, you must set up Datacap users for Datacap components for database authentication purposes.

For more information about database rights that need to be granted, see [Database security permissions](#).

Parent topic: [Domains and Windows accounts](#)

Authentication

You must determine which authentication system you are using and configure Datacap for user authentication or group authentication.

You must configure the Datacap accounts and groups that are based on your authentication system. For more information, see [Configuring the Datacap Server service to use an external authentication system](#).

Parent topic: [Domains and Windows accounts](#)

Stations

Station names in a Datacap application can be set up for each physical computer, for each type of workstation (such as Scan or Verify), or, for a group of computers in a department or a physical location.

You can limit the tasks that can be run by users that log in to a Datacap application with a specific station name. You can also route batches that are based on station name.

Depending on whether each station represents a single computer or a group of computers, you can monitor the activity on an individual computer or in aggregate.

Users that use interactive Datacap software components enter station names manually, and the station names they use do not need to match the name of the computer they are using. The background services or processes in each Datacap application provide the computer name automatically as the station name or must be configured to provide the station name.

For more information about using stations, see [Understanding virtual stations](#) and [Understanding queuing of batches to specific users and stations](#).

Parent topic: [Planning your Datacap system](#)

Databases

Datacap is installed with a Microsoft Access database, however, Microsoft Access is not supported in a Datacap production environment. You can configure your Datacap production environment to use a DB2®, Microsoft SQL Server, or Oracle database.

Review this planning information before you define and configure a DB2, Microsoft SQL Server, or Oracle database for your Datacap application databases.

You must be a Database Administrator with a working knowledge of the database management system to create a DB2, Microsoft SQL Server, or Oracle database. And to define the structure of the Datacap application databases for it.

The DB2, SQL Server, or Oracle database Server and Client software is not installed as part of the Datacap product installation. This software is available on the vendor media or by download from the vendor website.

Install the database server software on the computer where you want to run the database. You must have access to this computer.

Evaluate your needs and resources to decide the type of environment on which you want to install your database:

- In smaller environments, you can install the database server and some or all of the Datacap software components on the same computer
- In larger, distributed systems, you might want to install one database server for all of the clients that use the database. Or you can one database server for each environment, such as Test and Production.

You must run the database client software on every Datacap computer that accesses the database server.

Any computer that does lookups on the database must have access to the computer where the database server is installed.

Using the DB2, SQL Server, or Oracle Administration Tools, you create an initial, empty or database for the Administration, Engine, and Fingerprint databases. You must specify the elements of the database, for example the name of the database, user permissions.

Verify that the Datacap applications that you want to copy to the DB2, SQL Server, or Oracle were created.

- [Microsoft SQL Server prerequisites](#)
Before you can define and configure Datacap application databases for Microsoft SQL Server, ensure that the required software is installed on your system.
- [Oracle database prerequisites](#)
Before you can define and configure Datacap application databases for Oracle, ensure that the required software is installed on your system.
- [DB2 prerequisites](#)
Before you can define and configure Datacap application databases for DB2, ensure that the required software is installed on your system.
- [Connecting to databases](#)
You can use different authentication methods to provide access from Datacap to DB2, Microsoft SQL Server, or Oracle databases.

Parent topic: [Planning your Datacap system](#)

Related information:

[Datacap database configuration](#)

Microsoft SQL Server prerequisites

Before you can define and configure Datacap application databases for Microsoft SQL Server, ensure that the required software is installed on your system.

The following prerequisites must be installed and running on the appropriate computers in your Datacap system:

- Datacap 8.0 or later:
 - Set up Datacap Server Service

- Set up at least one Datacap Client station
- Microsoft SQL Server version 2005 or later on the database server computer
- Microsoft SQL Server 2008 Native Client on every Datacap computer that accesses the database server. The Native Client contains the SQL OLE DB provider driver that Datacap uses to connect to SQL Server. No further configuration is required.

Parent topic: [Databases](#)

Oracle database prerequisites

Before you can define and configure Datacap application databases for Oracle, ensure that the required software is installed on your system.

The following prerequisites must be installed and running on the appropriate computers in your Datacap system:

- Datacap 8.0 or later:
 - Set up Datacap Server Service
 - Set up at least one Datacap Client station
- Oracle Data Access Components (ODAC) on the database server computer. Create the Oracle database with UTF8 encoding to support Unicode.
- Oracle Data Access Components (ODAC) as the Oracle Client. The version of the client software must be 32-bit and match the version of your Oracle server.
 - Install Oracle Data Access Products software on the Datacap Server, any stations that run Maintenance Manager, and any Datacap Client station that requires a connection to the Fingerprint database of the application.
 - For Datacap applications that use fingerprints, install Oracle Data Access Products software on Rulerunner Service server stations, and development and management workstations that manage Fingerprints.
 - An Administration database installation of the Oracle Client includes all of the required components.
- Set up a Net Service Name for your Administration and Engine databases by using the Oracle Net Configuration Assistant. Set up the Net Service Name on the computer that hosts the Datacap Server. For a Fingerprint database, set up a Net Service Name on any Datacap Client station that requires a connection to the Fingerprint database of the application.

Parent topic: [Databases](#)

DB2 prerequisites

Before you can define and configure Datacap application databases for DB2®, ensure that the required software is installed on your system.

IBM® Datacap, release 9.0.1 is certified for IBM DB2 Workgroup Server Edition - Restricted Use V10.5 for Windows on AMD64 and Intel EM64T systems (x64) Multilingual (CIWN9ML)

The following prerequisites must be installed and running on the appropriate computers in your Datacap system:

- IBM Datacap Version 9.0.1:
 - Set up Datacap Server Service
 - Set up at least one Datacap Client station
- DB2 Server version 10.5 on the database server computer

- DB2 Client version 10.5 on the computers where the Application Copy Tool and the Datacap Server are running

The DB2 client libraries that you use for Datacap Version 9.0.1 can be found on the following page, under IBM Data Server Client: <http://www.ibm.com/support/docview.wss?uid=swg24038262>.

Select the version of Windows that matches your system.

Parent topic: [Databases](#)

Connecting to databases

You can use different authentication methods to provide access from Datacap to DB2®, Microsoft SQL Server, or Oracle databases.

About this task

You can connect to databases by using one of the following authentication methods:

- **Windows Authentication:** Uses Microsoft Active Directory Domain service account credentials for Datacap Server Service, Datacap Web, and Rulerunner Service. Create one or more local SQL Server user accounts, and grant them read/write access to the Datacap databases. When you create connection strings for your application to access the databases, include the appropriate local SQL Server user name and password in each connection string.
- **SQL Authentication:** Requires a SQL Server Login ID and password in connection strings that are provided by Datacap Application Manager.

For Oracle databases, you must use Oracle Authentication. This requires that each client or server provides an Oracle Login ID and password which must match the credentials of a user who is defined in the Oracle database.

Procedure

To connect to the database:

1. Configure Windows authentication for SQL Server:
 - a. Open the SQL Server Configuration Manager or SQL Server Management Studio and log in to Administrator or to an account with rights to create users.
 - b. Set up user credentials and permissions for the SQL Server. These user credentials and permissions must exist in the Datacap databases.
 - c. Grant Read/Write access to the Datacap databases to the Microsoft Active Directory Domain service account.
 - d. See the documentation that was provided with your SQL Server database for instructions on setting user credentials and permissions.
2. Configure SQL Server authentication:
 - a. Open the SQL Server Configuration Manager or SQL Server Management Studio and log in to Administrator or to an account with rights to create users.
 - b. Set up user credentials and permissions for the SQL Server. These user credentials and permissions must exist in the Datacap databases.
 - c. See the documentation that was provided with your SQL Server database for instructions on setting user credentials and permissions.
3. Configure Oracle authentication:
 - a. Open the Net Configuration Assistant and log in to Administrator or to an account with rights to create users.

- b. Create a single Oracle Net Service Name. A Net Service Name is required for every Datacap Server that accesses the Administration and Engine databases and every Datacap Client that accesses the Fingerprint database.
 - c. See the documentation that was provided with your Oracle database for instructions on creating a Net Service Name. When you create connection strings for Datacap to access the database, include the Oracle user name and password for the Datacap database in each connection string.
4. Configure DB2 authentication:

For DB2 databases, you must use DB2 Authentication. This requires that each client or server provides a DB2 login ID and password, which must match the credentials of a user who is defined in the DB2 database.

- a. Open the IBM® Data Studio Database Administration tool for DB2 10.5.
- b. Using the Administration Explorer, connect to the target DB2 database.
- c. See the documentation that was provided with your DB2 database for instructions on creating a user ID and password to use when logging into the database from a Datacap application..

When you create connection strings for Datacap to access the database, include the Datacap user name and password for the Datacap database in each connection string. If the user that created the database tables is different from the user ID that is connecting to the database, then you must also use the `CurrentSchema` setting in the connection string and specify the user ID that created the tables.

Parent topic: [Databases](#)

Installation methods

The installation plan for Datacap depends on whether you are upgrading from an earlier version, and whether you are installing to the default location on the machine, C:\Datacap.

You cannot install Datacap 9.0.1 on a machine on which any earlier versions of Datacap are installed. You can upgrade from an earlier version of Datacap, which involves installing Datacap 9.0.1 on machines that have no Datacap components, folders, registry keys, or websites already installed. Then, convert and migrate your applications from the earlier version to Datacap 9.0.1. For more information, see [Migrating from previous releases](#).

Although the installation package can be located anywhere (for example, on an accessible network drive or in a CD/DVD drive), the destination of the installed software must be on a physical hard disk drive of the machine from which you run Setup.exe.

Important:

You must install Datacap with all the required features. If you have a Datacap system with a limited set of features, then the installation of one or more new Datacap features on the same system might result in an installation error (error code 1334).

For example, if a system has only Taskmaster Client installed, then trying to install RuleRunner on the same system later might result in installation failure with error 1334.

The only characters that can be used in the Datacap 9.0.1 installation path are the alphanumeric (a-z, A-Z, 0-9), underscore (_), backslash (\), and colon (:) characters from the US ASCII character set. Other characters cause the installation process to fail.

You install Datacap to C:\Datacap by using the installation wizard.

Note: Applications are typically run from the directory where you installed the software, but it is not a requirement. For example, you might run your programs for testing from a C:\Test\ directory. However, if you try

to change the setup program for an application that is in a directory other than where you installed the Datacap software, the *.set files are not found when you change the setup program. An error occurs in the Datacap Web Client, such as `Unable to locate ProgramSet file 'rulerunner.exe.set.xml' at 'C:\Test\ProgramSet\Rulerunner.exe.set.xml'`. You can avoid this problem by copying the ProgramSet directory from the installation location to the directory that contains the application.

When you configure Datacap on a single workstation or in a client/server environment, your Windows user account must have administrator rights on the workstation.

Installing and configuring on a single workstation

You can install, configure, and run Datacap components on a single Microsoft Windows workstation. This configuration uses the Datacap TravelDocs sample application as the example so you can see it process batches of sample documents. Your Windows user account must have administrator rights on the workstation.

Installing and configuring in a client/server environment

You can install, configure, and run Datacap server and client components in a client/server environment. You can use the sample TravelDocs application to see how Datacap processes batches of sample documents. Your Windows user account must have administrator rights on every machine on which you install Datacap. For a checklist of the account and configuration settings, see *Client/server installation checklist*.

For information on how to obtain detailed hardware and operating system requirement reports for Datacap software, see: [System requirements](#).

Deploying Datacap servers and clients over WAN

Remote users that access Datacap over a wide area network (WAN) can use any of the following options:

- Thin clients based on Taskmaster Web
- Thin clients based on Datacap Navigator
- FastDoc Capture operating in offline mode

The Datacap thick clients (such as DotScan, DotEdit, or DcDesktop) and utilities (NENU, Fingerprint Maintenance Tool) require LAN communication speeds and low latency for responsive performance.

Connect all the Datacap Taskmaster Servers, Datacap Navigator servers, Rulerunner Servers, Web Servers, file servers, and databases to a single high performance LAN for best results. Network delays between Taskmaster Server, shared files and databases cause degraded performance of Job Monitor and data intensive operations.

Some customers successfully operate Datacap thick clients in remote sites using Citrix or other remote access technology.

Important:

IBM has not tested or sought certification with Citrix, and does not provide support for Citrix. If you deploy Datacap clients on Citrix and encounter issues, IBM might require you to reproduce the issues outside of Citrix as part of the investigation.

For more information about deployment recommendations and diagrams, see the IBM Redbook *Implementing Imaging Solutions with IBM Production Imaging Edition and IBM Datacap Capture*, section 2.5.

Parent topic: [Planning your Datacap system](#)

High availability (load balancing)

You can use network load balancers to manage client requests across servers in a Datacap system.

Load balancing is a method for scaling a system horizontally by distributing the work across many computer nodes in a "farm." It also provides high availability by redirecting clients to a working node in case of failure. A load balancer or content switch presents a single address for communication with multiple servers for one or more Datacap applications. You configure the load balancer to send requests that are directed to each pooled / balanced address to one of the servers in the farm. You can select round-robin scheduling or another method. Configure the Datacap Server connection timeout to be longer than the processing time for the longest batch. Typically, 1 hour is sufficient.

If possible, you can grant network access to the back-end server addresses for easier initial setup and subsequent problem solving. Test your system without any load balancing at first. Add load balancing to one component at a time, and reconfigure as needed. Test each balanced address, including failover to each back-end server, before you test the next component. If policy requires that you disable connections to the back-end servers, be prepared to re-enable for troubleshooting, if required.

Datacap Server

Clients access the Datacap Server by using a TCP/IP socket-based protocol. You configure the Datacap Server name or IP address and port in Datacap Application Manager. The Datacap Server normally listens on port 2402, but you can change the port number in Datacap Server Manager. If you change the port number in Datacap Server Manager, you must also configure the port number in Datacap Application Manager. If a load balanced server fails, all new client requests are directed to a different server, the old session becomes invalid, and the user or client must log in again. Any outstanding server requests are terminated and batches that are in process via that server remain in a running status. Users who logged in to this server receive an error message and must log in again. TCP/IP sessions are inherently persistent and there is no need for the load balancer to persist Datacap Server sessions. However, you might want to configure the load balancer to persist sessions based on the client IP address to force all threads from any Rulerunner server to use the same Datacap Server. For high availability, the best practice is to configure multiple Datacap Server instances in a farm by using a network load balancer or content switch. Enter the single virtual (balanced) address for Datacap Server in Datacap Application Manager. Configure the load balancer with persistent, sticky sessions based on the source IP.

Datacap Web Client

The servers where the Datacap Web Client is located can be farmed. Designate one or more IP addresses or port on your network for your Datacap Web Client site home page. Client browsers connect to this load balancer port by using an HTTP or HTTPS protocol. Configure your load balancer to redirect those requests to individual web servers by using round-robin scheduling, or another method. Ports 80 and 443 are standard but you can configure an alternate port in the Microsoft IIS Manager. The Datacap Web Client uses session cookies, so you must configure the load balancer to persist sessions based on the client's IP address. Set the load balancer session timeout to match the IIS session timeout. If a server fails, users who connected to the failed server receive an error message and must log in again.

Datacap Web Services

Datacap Web Services servers can be farmed. Clients connect to an address and port for the load balancer and are directed to a specific server. Sessions must be persisted by the load balancer and the session timeout must match the web service session timeout. There are different methods to achieve session persistence, refer to your specific load balancer manual for complete details. Failure of a Datacap Web Services server generates an error for requests that are in progress and the requested operations might not be completed.

Datacap Report Viewer

The Datacap Report Viewer IIS servers can be farmed. Clients connect to an address and port for the load balancer and are directed to a specific server. Sessions must be persisted based on IP address, and the session timeout must match the IIS server session timeout. If a Datacap Report Viewer server fails, all existing sessions end.

Fingerprint Service

The Fingerprint Service servers can be farmed, if the fingerprints are static during normal system operation. Updates and deletions of fingerprints are not synchronized automatically between servers. Fingerprint servers must be restarted, or their contents programmatically reset to keep them synchronized, if changes are made to the set of fingerprints. You must configure the load balancer to persist sessions based on the client's IP address.

Rulerunner

The Rulerunner servers independently poll Datacap servers for pending work. These servers do not require or benefit from load balancing.

Datacap Navigator

Datacap Navigator is a plug-in for IBM® Content Navigator. For information about configuring a load-balanced environment, see [Getting IBM Content Navigator up and running](#). Refer to the steps that are marked High Availability Clusters.

Table 1. Load balancing options for Datacap servers

Datacap Server	Load Balanced	Protocol
Datacap Server (application server)	Yes, persistent sessions by client IP	TCP/IP socket
Datacap Web Client server	Yes, persistent sessions by client IP	HTTP, HTTPS
Datacap Web Services server	Yes, persistent sessions by client IP	HTTP, HTTPS
Datacap Report Viewer server	Yes, persistent sessions by client IP	HTTP, HTTPS
Fingerprint Service server	Yes, persistent sessions by client IP	HTTP, HTTPS
Rulerunner server	No	

Parent topic: [Planning your Datacap system](#)

Prerequisites for installing Datacap

Before you begin any installation of Datacap, ensure that your system is configured with the required prerequisite software.

For more information, see [System requirements](#).

Attention: Datacap Insight Edition requires a 32-bit Java runtime environment (JRE). The default location for the JRE is *Datacap_Installation\dcshared\jre*. If you install the JRE in another location, you must specify the installation path in the `JAVA_HOME` system environment variable.

- [Backing up and removing Datacap version 8.0.1 or earlier](#)
You cannot install Datacap 9.0.1 on a machine on which any earlier versions of Datacap are installed.
- [Installation and configuration prerequisites](#)
Before you begin installing Datacap software, ensure that you meet these prerequisites. These prerequisites ensure that you can install, configure, and run the Datacap components of an application either on a single machine, or on multiple machines in a client/server distributed network configuration.
- [FastDoc prerequisites](#)
Before you install FastDoc on a stand-alone workstation, ensure the required software is installed on your system.
- [Microsoft Internet Information Services and Microsoft .NET Framework](#)
If you are going to install one or more of the Datacap Web Client server components, Report Viewer, Fingerprint Service, or Datacap Web Services, Microsoft IIS must be installed before Microsoft .NET Framework 3.5.1. If IIS is installed after .NET,, you must run the Microsoft ASP.NET IIS Registration Tool (aspnet_regiis.exe) with the installation parameter -i to update the ASP.NET application script maps to point to the appropriate ASP.NET ISAPI version.
- [Turning on Microsoft .NET Framework 3.5.1](#)
Before you install Datacap on your Windows workstations or servers, ensure that the Microsoft .NET Framework 3.5.1 operating system feature is turned on or added on every workstation and server.
- [Updating ASP.NET](#)
Follow these instructions when you know that Microsoft .NET was installed before Microsoft Internet Information Services (IIS) on the web server, or if you are not sure that Microsoft Internet Information Services was installed first.
- [Verifying that IIS components are installed](#)
Ensure that the IIS Web Server Role Services are installed for the Datacap Web Client server components, such as Datacap Web Client, Report Viewer, Fingerprint Service, and Datacap Web Services.
- [Setting up the scanner](#)
When a computer is to be used as a scan station, follow the manufacturer's instructions to attach the scanner to the computer. Install and configure the required scanner drivers and software. Be sure you can scan successfully with an image capture software product other than Datacap. If you are unable to scan documents with software that is not Datacap, you cannot scan documents with Datacap.
- [Ensuring the correct Windows language is selected](#)
When you process images with data in a language other than English on an English language version of Windows, ensure that the Windows Language for non-Unicode programs setting matches the language that you want to recognize.
- [Installing Microsoft .NET Framework 4.0](#)
Computers running Datacap software components require Microsoft .NET Framework 4.0. If Microsoft .NET Framework 4.0 is not already installed, it is installed automatically during the Datacap software installation process.

Parent topic: [Installing](#)

Backing up and removing Datacap version 8.0.1 or earlier

You cannot install Datacap 9.0.1 on a machine on which any earlier versions of Datacap are installed.

About this task

To upgrade from an earlier version of Datacap, install Datacap 9.0.1 on machines on which no Datacap components, folders, registry keys, or websites exist. You must convert and migrate your applications from the earlier version to Datacap 9.0.1. For more information, see [Migrating from previous releases](#).

If you are going to install Datacap 9.0.1 on a machine that was part of an earlier Datacap system, you must remove all traces of Datacap before you begin.

Parent topic: [Prerequisites for installing Datacap](#)

Installation and configuration prerequisites

Before you begin installing Datacap software, ensure that you meet these prerequisites. These prerequisites ensure that you can install, configure, and run the Datacap components of an application either on a single machine, or on multiple machines in a client/server distributed network configuration.

Datacap Web Client server prerequisites

Before you install the Datacap Web Client server components on a Windows server, add the Microsoft IIS Web Server Role using the Add role wizard.

After you add the IIS Web Server Role, add the .NET 3.5.1 Feature.

If you are not sure whether these components are installed on your web servers, or if you are not sure that they were installed in the correct sequence, see the following.

- [Microsoft Internet Information Services and Microsoft .NET Framework](#)
- [Updating ASP.NET](#).

Datacap server and workstation prerequisites

Before you install any Datacap components, ensure that Microsoft .NET 3.5.1 is installed on all servers and workstations on which Datacap is to be installed. For instructions, see [Turning on Microsoft .NET Framework 3.5.1](#).

Datacap components that are installed in a multi-machine environment use TCP port 2402 to access the Datacap Server. If there is a firewall on the Datacap Server, you must open this port.

Datacap software component dependencies

Some Datacap software components depend on the presence of other Datacap software components to run.

- To export to Microsoft SharePoint using FastDoc, the Datacap for Microsoft SharePoint connector is required.
- To process Adobe PDF documents using Flex, the Datacap for email and Electronic Documents connector is required.

Parent topic: [Prerequisites for installing Datacap](#)

FastDoc prerequisites

Before you install FastDoc on a stand-alone workstation, ensure the required software is installed on your system.

The following prerequisites must be met before you can install FastDoc:

- If you are processing images and capturing data in a non-English language, make sure the correct Windows language is selected.
- For Windows 7 machines, ensure that Microsoft .NET Framework 4.5.1 is installed.

- FastDoc requires additional Microsoft software that, if it is not already installed, is installed automatically during the installation process.
- Ensure that your scanner is set up and works outside of FastDoc.
- If you are processing pre-scanned images from other scanners, ensure all images are scanned at the resolution that you are using in FastDoc and ensure that the resolution is not changed over time. Resolutions 300 DPI or 78 DPCM are recommended.

Parent topic: [Prerequisites for installing Datacap](#)

Related tasks:

[Installing FastDoc on one machine](#)

Microsoft Internet Information Services and Microsoft .NET Framework

If you are going to install one or more of the Datacap Web Client server components, Report Viewer, Fingerprint Service, or Datacap Web Services, Microsoft IIS must be installed before Microsoft .NET Framework 3.5.1. If IIS is installed after .NET, you must run the Microsoft ASP.NET IIS Registration Tool (aspnet_regiis.exe) with the installation parameter -i to update the ASP.NET application script maps to point to the appropriate ASP.NET ISAPI version.

About this task

For instructions on running the tool, see [Updating ASP.NET](#). For more information about the tool, search for aspnet_regiis.exe on the Microsoft website.

Procedure

To determine whether IIS and .NET are already installed.

1. From Start, select Administrative Tools > Server Manager.
2. In the Server Manager hierarchy pane, expand Roles. The Web Server (IIS) role is visible.
3. To determine whether the Microsoft .NET Framework 3.5.1 is installed, expand the Features Summary. An entry for the Microsoft .NET Framework 3.5.1 feature is displayed.
4. Depending on whether either or both are installed, do one of the following.
 - a. If .NET and IIS are not displayed, install IIS, then install Microsoft .NET Framework 3.5.1. Then, continue with the Datacap installation process.
 - b. If both .NET and IIS are displayed, run aspnet_regiis.exe, continue with the Datacap installation process.
 - c. If only IIS is displayed, install Microsoft .NET Framework 3.5.1. Then, continue with the Datacap installation process.
 - d. If only .NET is displayed, install IIS, then run aspnet_regiis.exe. Then, continue with the Datacap installation process.

Parent topic: [Prerequisites for installing Datacap](#)

Turning on Microsoft .NET Framework 3.5.1

Before you install Datacap on your Windows workstations or servers, ensure that the Microsoft .NET Framework 3.5.1 operating system feature is turned on or added on every workstation and server.

Procedure

1. To turn on Microsoft .NET Framework 3.5.1 on Windows 7:
 - a. Select Start > Control Panel > Programs > Programs and Features.
 - b. Select Turn Windows features on or off.
 - c. If not already installed, select Microsoft .NET Framework 3.5.1 and click OK.
 - d. Restart the machine when prompted.
2. To turn on Microsoft .NET Framework 3.5.1 on Windows 2008:
 - a. Select Start > Administrative Tools > Server Manager.
 - b. In the Server Manager hierarchy pane, select Features.
 - c. If not already installed, click Add Features.
 - d. Select .NET Framework 3.5.1 Feature and click Next.
 - e. Click Install.
 - f. Ensure that the installation was successful, then click Close.

Parent topic: [Prerequisites for installing Datacap](#)

Updating ASP.NET

Follow these instructions when you know that Microsoft .NET was installed before Microsoft Internet Information Services (IIS) on the web server, or if you are not sure that Microsoft Internet Information Services was installed first.

Before you begin

In Windows 2008, Microsoft aspnet_regiis.exe must be run in Administrator mode, and can be run in a command prompt window. Before you begin, ensure that a shortcut exists, or add a shortcut to %windir%\system32\cmd.exe to your Start menu.

About this task

This procedure provides instructions on how to run Microsoft aspnet_regiis.exe to update the ASP.NET application script maps to point to the appropriate version of ASP.NET.

- When the server hosts Datacap Web Client, Report Viewer, or Datacap web services (wTM), run the aspnet_regiis.exe found in the C:\Windows\Microsoft.NET\Framework\v4.0.30319\ folder.
- When the server hosts the Fingerprint Service, run the aspnet_regiis.exe program file that is found in the C:\Windows\Microsoft.NET\Framework\v2.0.50727\ folder.
- If the server hosts Datacap Web Client, Report Viewer, or Datacap web services (wTM), and the Fingerprint Service, run both versions of aspnet_regiis.exe.

Procedure

1. On the Start menu, right-click the shortcut for the command prompt window and select Run as administrator.
2. Run the aspnet_regiis.exe with the -i parameter, using the entire path, such as in the following examples: C:\>C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis.exe -i
And C:\>C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis.exe -i.
Progress messages are displayed indicating the start and completion of the installation.
3. When the installation is finished, close the command prompt window.

Parent topic: [Prerequisites for installing Datacap](#)

Verifying that IIS components are installed

Ensure that the IIS Web Server Role Services are installed for the Datacap Web Client server components, such as Datacap Web Client, Report Viewer, Fingerprint Service, and Datacap Web Services.

About this task

The Datacap Web Client server components must have specific IIS Web Server Role Services installed.

Procedure

Ensure that the IIS components are installed.

1. From the Start menu on the web server, select Administrative Tools > Server Manager.
2. In the Server Manager hierarchy pane, expand Roles and select Web Server (IIS).
3. In the Web Server (IIS) pane, expand Role Services. Under Common HTTP Features ensure that Static Content, Default Document, Directory Browsing, and HTTP Errors are installed. If you are installing wTM, do not install the WebDAV Publishing role service because it prevents the Datacap web services PUT method from functioning.
4. In the Role Services pane under Application Development, ensure that the ASP.NET, .NET Extensibility, ASP, ISAPI Extensions, and ISAPI Filters are installed.
5. In the Role Services pane, under Health and Diagnostics ensure that HTTP Logging and Request Monitor are installed.
6. In the Role Services pane, under Security ensure that Request Filtering is installed.
7. In the Role Services pane, under Performance ensure that Static Content Compression is installed.
8. In the Role Services pane, under Management Tools ensure that IIS Management Console is installed.
9. (Windows Server 2012 R2) In Server Manager, click Dashboard, and click Add roles and features.
 - a. In the Add Roles and Features Wizard, go to the Features section.
 - b. In .NET Framework 4.5 Features > WCF Services, select the HTTP Activation check box.
10. Close the Server Manager window.

Parent topic: [Prerequisites for installing Datacap](#)

Setting up the scanner

When a computer is to be used as a scan station, follow the manufacturer's instructions to attach the scanner to the computer. Install and configure the required scanner drivers and software. Be sure you can scan successfully with an image capture software product other than Datacap. If you are unable to scan documents with software that is not Datacap, you cannot scan documents with Datacap.

About this task

Tip: A physical scanner is not required for running the Datacap that is separately licensed or sample applications because those applications can process the prescanned, sample images that are included during installation.

Parent topic: [Prerequisites for installing Datacap](#)

Ensuring the correct Windows language is selected

When you process images with data in a language other than English on an English language version of Windows, ensure that the Windows Language for non-Unicode programs setting matches the language that you want to recognize.

Procedure

To ensure that the correct language has been selected in Windows.

1. From the Windows Start menu, select Control Panel. The Control Panel window opens.
2. Double-click Regional and Language. The Regional and Language window opens.
3. Click the Administrative tab.
4. Click Change system locale. The Regional and Language Settings window opens with the Current® language for non-Unicode programs displayed.
5. If the correct language is displayed, click Cancel to close the window.
6. If the language displayed does not match the language that is to be recognized, select the correct locale, click OK and follow the prompts to restart the workstation.

Parent topic: [Prerequisites for installing Datacap](#)

Related tasks:

[Installing FastDoc on one machine](#)

Installing Microsoft .NET Framework 4.0

Computers running Datacap software components require Microsoft .NET Framework 4.0. If Microsoft .NET Framework 4.0 is not already installed, it is installed automatically during the Datacap software installation process.

About this task

You can install Microsoft .NET Framework 4.0 on your servers and workstations before you install Datacap to shorten the time that is required for the Datacap installation.

Parent topic: [Prerequisites for installing Datacap](#)

Ensuring that the Datacap Server Service is started

The Datacap Server Service must be running before you start an application or a Datacap component. You can check the status of the service and start the service in the Datacap Server Manager.

About this task

All configurations of Datacap require that the Datacap Server Service must be running before you can successfully start an application or Datacap component. This requirement applies whether the Datacap Server Service is started on the Workstation in a single computer configuration or on the Datacap Server for client/server configurations.

In a single computer configuration, the Datacap Server Service runs by using the LocalSystem account.

In a client/server configuration, the Datacap Server Service runs by using the account that you set up for it.

You start the Datacap Server Service by using the Datacap Server Manager. You can also ensure that the Datacap Server Service is started by checking the status message that is displayed by the Datacap Server Manager.

Procedure

To start the Datacap Server Service:

1. From Start, select IBM Datacap Services > Datacap Server Manager. If the User Account Control window opens, click Yes.
2. Check the Status message and start the service if it is stopped.
Tip: If a Datacap Service error displays when you start the service in a client/server environment, the Service does not have adequate rights to log on to the server. For more information, see *Granting Datacap Server Service the Log On as a Service Privilege*.
3. Close the Datacap Server Manager window.

Related tasks:

[Granting Datacap server service the Log On as a Service privilege](#)

Ensuring that the Datacap Server Service is stopped

You can stop the Datacap Server Service in the Datacap Server Manager to do maintenance tasks and other activities. You can also ensure that the Datacap Server Service is stopped by checking the status message that is displayed by the Datacap Server Manager.

About this task

If you are stopping the Datacap Server Service in a client/service environment, run these steps on the server where the Datacap Server is installed.

Procedure

To stop the Datacap Server Service:

1. From the Windows Start menu, select IBM Datacap Services > Datacap Server Manager. When User Access Control (UAC) is enabled, the User Account Control window is displayed. Click Yes.
2. Check the Status message and stop the service if it is started.
3. Close the Datacap Server Manager window.

Installing and configuring Datacap on one machine

You can install, configure, and operate Datacap on a single machine for demonstration or testing purposes only.

Datacap is designed to be installed, configured, and operated in a client/server environment. The following components can be installed, configured, and run on the single Windows machine for demonstration or testing purposes.

- Datacap Server service
- Datacap Web Client
- Report Viewer
- Rulerunner Service (single thread)

You must configure the single machine for Datacap authentication. The Datacap Server service, Datacap Web Client application pool, Report Viewer application pool, and Rulerunner Service use the default LocalSystem account on the single machine.

The Windows account that you use when you log in to the single machine must have administrator rights.

After you install the Datacap software from the installation wizard, you must apply the software license by going to the IBM Datacap Services folder, opening the Datacap License Manager, and selecting your license options. The IBM® Datacap *.swidtag license files are in the C:\Datacap\iso-swid folder.

- [Installing FastDoc on one machine](#)
You run the installation program wizard to install the FastDoc components on a single workstation where the Datacap software is not installed.
- [Single machine installation and configuration](#)
You can install, configure, and operate Datacap on a single machine for demonstration or testing purposes only.

Parent topic: [Installing](#)

Installing FastDoc on one machine

You run the installation program wizard to install the FastDoc components on a single workstation where the Datacap software is not installed.

About this task

You can get the installation package, Setup.exe, from a network drive, local drive, or a CD/DVD drive. You must install FastDoc on the physical hard disk of the computer from which you run Setup.exe.

For security purposes, the Key Management tool (dcskey.exe) is installed on the Datacap Server. You can use dcskey.exe to generate the security encryption key on FastDoc based on the following installation use cases. Tip: In the use cases where FastDoc does not use Datacap Server, the server still must be installed for FastDoc to be able to obtain dcskey.exe.

Table 1. Installation use cases

Use case	Action required
FastDoc runs as a client to Datacap Server	Import dcskey.exe from the server
FastDoc is not a Datacap client, shares Datacap Application Manager files with other FastDoc clients	Generate the key on one of the clients and import it to the other clients
FastDoc is not a Datacap client, which is used by one user on a stand-alone workstation and does not share Datacap Application Manager files with other clients	None

After you generate the key on the FastDoc client, delete dcskey.exe and the key file from the client computer.

Procedure

To install FastDoc on a stand-alone workstation:

1. Place the installation package on your network.
2. Open Windows Explorer, go to and double-click the Setup.exe file. If the User Account Control window displays, click Yes.
3. At the Select the language dialog, select the appropriate language and click OK. The language that you select determines the language that is displayed by the installation program wizard during the installation process.
4. At the Welcome window, click Next.
5. At the Software License Agreement window, select I accept the terms in the license agreement and click Next.
6. At the Setup Type window, select Custom and click Next.
7. At the Custom Setup window, expand the Datacap client component and select FastDoc if it is not already selected.

8. Click all of the other components and select X This feature will not be available and click Next. A red X is displayed next to these components.
9. At the Ready to install the Program window, click Install. FastDoc and the other standard components are installed.
10. Click Finish to close the installation program wizard.

Parent topic: [Installing and configuring Datacap on one machine](#)

Related concepts:

[FastDoc prerequisites](#)

Related tasks:

[Ensuring the correct Windows language is selected](#)

Installing Datacap on one machine

You must follow a specific procedure to install the Datacap components on a Windows single machine. The configuration of Datacap on a single machine is for demonstration or test purposes only. You can install the TravelDocs application to confirm that the software is installed and configured correctly.

About this task

The Datacap installation and configuration steps are performed in the following sequence.

Procedure

1. Ensure that you complete the prerequisites that are detailed in *Installation and Configuration Prerequisites*.
 2. Add the local system account to the single machine's Administrators group.
 3. Run the Datacap installation program, install the Datacap software components and any additional components for which you have a license on the single machine.
 4. Export the encryption keys.
 5. Start the Datacap server service.
 6. Run the Datacap TravelDocs application.
 7. Stop the Datacap server service.
 8. Configure Datacap Web Client.
 9. Configure Internet Explorer.
 10. Start the Datacap server service.
 11. Run the Datacap TravelDocs application.
 12. Configure Report Viewer and view reports.
 13. Configure the Rulerunner Service to run tasks on a single thread.
 14. Run TravelDocs tasks.
- [Adding the local system account to Administrators group](#)
Add the local system account of the workstation to the Administrators group of the workstation. The account now has local administrator rights on the computer.
 - [Installing Datacap components on one machine](#)
The procedure to install Datacap components on a single machine includes installing the required components and any separately licensed applications and connectors.
 - [Exporting encryption keys](#)
You must generate security keys that allow Datacap to encrypt and decrypt passwords. To replace existing keys with new keys, you can specify a parameter to generate encryption keys to a local store. You can specify a different parameter to export the encryption keys from the local keystore to a file that can be imported to other computers.

- [Datacap Web Client installation and configuration](#)
You can run tasks from a computer on which the Internet Explorer browser is installed when you set up the Datacap Web Client. In a single machine configuration that is used for testing and demonstration purposes, Datacap Web Client runs using the LocalSystem account.
- [Datacap Report Viewer](#)
The Datacap Report Viewer web application displays real-time reports of activity that is related to your Datacap applications.
- [Rulerunner Service \(single thread\)](#)
The Rulerunner Service is a Datacap component that runs Datacap Studio application background tasks that require no human interaction. Examples of background tasks are VScan, recognition, image pre-processing, validation, and export tasks.

Related concepts:

[Installation and configuration prerequisites](#)

Adding the local system account to Administrators group

Add the local system account of the workstation to the Administrators group of the workstation. The account now has local administrator rights on the computer.

Parent topic: [Installing Datacap on one machine](#)

Installing Datacap components on one machine

The procedure to install Datacap components on a single machine includes installing the required components and any separately licensed applications and connectors.

About this task

Run the Datacap installation program wizard to install Datacap on a single machine. The components that you install include the main Datacap components and any separately licensed applications and connectors.

These instructions apply to single machines with a Windows operating system.

Procedure

To install Datacap on a single machine:

1. Make the installation package available on your network or insert the Datacap CD in the Workstation's CD/DVD drive. If the installation process does not start automatically or if the package is on the network, open Windows Explorer, navigate to and double-click the Setup.exe. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.
2. Select the appropriate language, then click OK. The language that you select determines the language that is displayed by the installation program wizard during the installation process.
3. When more software is needed, the installation program wizard displays a list of the items to be installed. Click Install.
4. Click Next.
5. Accept the license agreement and click Next.
6. Select the Custom option. Click Next.
7. Include all of the Datacap software components.
8. Expand the Datacap client software component and include only the separately licensed applications.
9. Expand the Connectors software component and include only the connectors for which you have a license.

10. Click Next.
11. Click Install.
12. Click Finish.

Parent topic: [Installing Datacap on one machine](#)

Exporting encryption keys

You must generate security keys that allow Datacap to encrypt and decrypt passwords. To replace existing keys with new keys, you can specify a parameter to generate encryption keys to a local store. You can specify a different parameter to export the encryption keys from the local keystore to a file that can be imported to other computers.

About this task

You must generate and use the security encryption keys that allow Datacap to encrypt and decrypt the passwords that are used by users, services, and processes to access the Datacap server service and to log in to databases.

In a single machine configuration, you must generate and export the encryption keys that all of the Datacap components on the single machine use.

In a client/server configuration, you must generate and export matching security encryption keys from the server on which the Datacap server software component is installed to all of the computers on which any Datacap component is installed. This requirement secures any passwords that are passed over or received from the network by the Datacap component.

Procedure

To generate encryption keys and export them:

1. Open a command prompt and navigate to the C:\Datacap\Taskmaster folder. In a client/server configuration, perform this step on the computer on which the Datacap server software component is installed.
2. Run the key management program, `dcscopy.exe`, inserting one or more of the following parameters in the command. For example, to export keys during a new Datacap installation, you would enter `dcscopy.exe`

`e`

Exports the encryption keys from the local keystore to a `dc_KTF.xml` key transport file. You can use this file to import the keys to other computers. If no keys exist in the keystore, the `e` parameter generates new ones before the export. If keys exist in the keystore, the `e` parameter exports those keys.

`gnk`

Generates, but does not export, encryption keys in the local keystore. Use this parameter any time you must replace existing keys with new keys. For example, you would run the command `dcscopy.exe gnk e` to replace existing keys and export them. The newly exported keys would then must be imported onto all other Datacap computers in your configuration.

Parent topic: [Installing Datacap on one machine](#)

Datacap Web Client installation and configuration

You can run tasks from a computer on which the Internet Explorer browser is installed when you set up the Datacap Web Client. In a single machine configuration that is used for testing and demonstration purposes, Datacap Web Client runs using the LocalSystem account.

The steps that you must follow to install and configure Datacap Web Client include ensuring that the required IIS components are installed and creating the Datacap Web Client site. You use the Datacap Web Client Server Configuration tool to set up the Datacap Web Client site for a single machine configuration. A single machine configuration requires the tmweb.net application pool to be set for the LocalSystem account.

- [Creating the Datacap Web Client site](#)
You can create the Datacap Web Client site to run tasks from a computer on which the Internet Explorer browser is installed. The steps that you follow to install and configure the Datacap Web Client site include ensuring that the required IIS components are installed and configuring the tmweb.net application pool.
- [Internet Explorer configuration for Datacap Web Client](#)
You must configure Internet Explorer to run Datacap Web Client in a single machine installation.

Parent topic: [Installing Datacap on one machine](#)

Creating the Datacap Web Client site

You can create the Datacap Web Client site to run tasks from a computer on which the Internet Explorer browser is installed. The steps that you follow to install and configure the Datacap Web Client site include ensuring that the required IIS components are installed and configuring the tmweb.net application pool.

Before you begin

Before creating the Datacap Web Client site, [stop or ensure the Datacap Server Service](#) is stopped.

About this task

You must create the Datacap Web Client site by using the Datacap Web Client Server Configuration tool.

Procedure

1. From the Windows Start menu, select IBM Datacap Web > Datacap Web > Server Configuration Tools. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.
2. Determine whether the contents of the information box indicate one of the following conditions:
 - If one or more of the items that are listed in the information box are Not Found, click OK, then follow the instructions in [Microsoft Internet Information Services and Microsoft .NET Framework](#) to determine whether IIS and .NET were installed in the correct sequence and that all of the IIS components are installed.
 - If all of the items that are listed in the information box are Found, click OK and continue with these instructions.
3. Ensure that the default value for each of the settings is appropriate for your web site, processing needs, and corporate requirements.
 - When your Datacap Web Client site is not nested under the Default Web Site, select the appropriate Site and change the Site Settings.
 - Depending on the usual size of the images or groups of images that are uploaded from the Datacap Web Client, you must adjust the value of the ASP.NET Maximum File Upload Size and ASP Maximum Requesting Entity Body Limit fields.
 - Depending on your typical processing loads and schedules, change the default value of the App Pool Recycling Schedule to a time during which there are few or no batches in process Datacap

Web Client. For more information on recycling and Datacap Web Client, see *Effects of IIS application pool recycling on Datacap Web Client batches*.

- o Depending on your company's security policies, change the value of the Connection time-out field.
4. Click Configure.
 5. Click OK, then click Exit. The Datacap Web Client site is created.
 6. From the Windows Start menu, select Control Panel > Administrative Tools, then double-click Internet Information Services (IIS) Manager.
 7. In the Connections pane, expand the computer node, expand the Sites node, and expand the Default Web Site or your web site. The tmweb.net site is displayed. If it is not displayed, right-click the site and select Refresh.
 8. In the Application Pools pane, select the tmweb.net application pool, then in the Actions pane, in the Edit Application Pool section, click Advanced Settings.
 9. In the Process Model section, click the browse button to the right of Identity.
 10. In the Application Pool Identity window, change the Built-in account to LocalSystem, then click OK.
 11. In the Process Model section, set Load User Profile to True.
 12. Click OK.
 13. In the Connections pane, expand the Sites node, and expand the Default Web Site or your web site.
 14. Select the tmweb.net site, and in the middle pane, double-click Session State.
 15. Under Cookie Settings, change the Name to tmweb or another unique name, then, in the Actions pane, click Apply.
 16. In the Connections pane, select the Default Web Site, then, in the Actions pane, under Manage Web Site, click Restart.
 17. In the Connections pane, select the Default Web Site or your web site, then, in the Actions pane, under Manage Web Site, click Restart.

Parent topic: [Datacap Web Client installation and configuration](#)

Related information:

[Effects of application pool recycling on Datacap Web Client batches](#)

Internet Explorer configuration for Datacap Web Client

You must configure Internet Explorer to run Datacap Web Client in a single machine installation.

You can configure Internet Explorer to run locally on the same computer as Datacap Web Client in a single machine installation for demonstration and testing purposes. The procedure for configuring Internet Explorer includes adding the TMWeb.net address as a trusted site and testing Internet Explorer using the Datacap Web Client Configuration tool.

- [Adding the tmweb.net address as a trusted site](#)
You must add the tmweb.net address as a trusted site in a single machine installation to prevent Internet Explorer from blocking access to the Datacap Web Client site.
- [Configuring Internet Explorer on a single machine](#)
You can configure Internet Explorer to run on a single machine with the Datacap Web Client for demonstration and test purposes.

Parent topic: [Datacap Web Client installation and configuration](#)

Adding the tmweb.net address as a trusted site

You must add the tmweb.net address as a trusted site in a single machine installation to prevent Internet Explorer from blocking access to the Datacap Web Client site.

About this task

In a single machine installation, you can enter the default server address as a trusted site for Internet Explorer access to the Datacap Web Client site.

Procedure

1. Open Internet Explorer.
2. On the Tools menu, select Internet Options to open the Internet Options dialog.
3. Click the Security tab to display the options.
4. Select the Trusted sites option and press Sites to open the Trusted Sites dialog.
5. Enter the default server address `http://localhost` in the Add this website to the zone field, then click Add.
6. Close the Trusted Sites dialog and select OK to close the Internet Options dialog.

Parent topic: [Internet Explorer configuration for Datacap Web Client](#)

Configuring Internet Explorer on a single machine

You can configure Internet Explorer to run on a single machine with the Datacap Web Client for demonstration and test purposes.

About this task

This procedure provides instructions on how to set up Internet Explorer using the Datacap Web Client Configuration tool in a single machine configuration. When you configure Internet Explorer, you can use the Datacap Web Client for processing tasks. This procedure also provides instructions on how to test your configuration of Internet Explorer.

Important: When the operating system of the machine on which you want to run Internet Explorer to access Datacap Web Client is a 64-bit operating system, you must use the 32-bit version of Internet Explorer to access Datacap Web Client

Procedure

1. From the Windows Start menu, select IBM Datacap Web > Datacap Web Client Configuration Tools. When User Access Control is on, the User Account Control window opens. Click Yes.
2. Ensure that `http://localhost` is the default URL that is displayed.
3. Click Configure, OK, and Exit.
4. Start Internet Explorer and enter the URL for Datacap Web Client, followed by the `tmweb.net` virtual directory and the test page, `http://localhost/tmweb.net/ietest.aspx`.
5. When User Access Control is on, the first time you access the test page the User Access Control window opens. Click Yes to download the Datacap TIFF Viewer, Thumbnails, and DataEdit Controls.
6. Click Test. The green check marks are displayed when the test completes successfully. If the Datacap TIFF Viewer does not download, switch to the 32-bit version of Internet Explorer and repeat this procedure from step 1.

Parent topic: [Internet Explorer configuration for Datacap Web Client](#)

Datacap Report Viewer

The Datacap Report Viewer web application displays real-time reports of activity that is related to your Datacap applications.

For a list of the standard reports, see [Standard reports](#).

- [Overview of Datacap Report Viewer installation and configuration](#)
You must complete the required tasks to install and configure the Datacap Report Viewer reporting component.
- [Adding an application pool for Report Viewer](#)
When you are running on Windows 7 or Windows 2008, you must add a Microsoft Internet Information Services (IIS) Application Pool for use by Report Viewer.
- [Single machine installation: Setting up the Datacap Report Viewer website](#)
You must set up the Report Viewer website on Microsoft Internet Information Services (IIS) 7.5.
- [Adding the Datacap Report Viewer address as a trusted site](#)
You add the Report Viewer address as a trusted site in a single computer installation of Datacap to prevent Internet Explorer from blocking access to the Datacap website.

Parent topic: [Installing Datacap on one machine](#)

Related information:

[Custom Datacap Report Viewer reports](#)

Overview of Datacap Report Viewer installation and configuration

You must complete the required tasks to install and configure the Datacap Report Viewer reporting component.

Before you start, ensure that you met the installation and configuration prerequisites.

In a single computer configuration, Report Viewer runs by using the LocalSystem account. To set up a single computer configuration:

1. Ensure that you have the required Microsoft Internet Information Services components installed.
2. Add an Application Pool for Report Viewer.
3. Set up the Report Viewer website.

In a client/server environment:

1. Create or ensure that an account exists for Report Viewer.
2. Ensure that you have the required Microsoft Internet Information Services (IIS) components installed.
3. Set up various Report Viewer permissions.
4. Install Report Viewer on the web server.
5. Import encryption keys on the web server.
6. Add an Application Pool for Report Viewer.
7. Set up the Report Viewer website and Application Pool advanced settings.

Parent topic: [Datacap Report Viewer](#)

Related concepts:

[Installation and configuration prerequisites](#)

Adding an application pool for Report Viewer

When you are running on Windows 7 or Windows 2008, you must add a Microsoft Internet Information Services (IIS) Application Pool for use by Report Viewer.

Procedure

To add an Application Pool for Report Viewer:

1. From the Windows Start menu, select Administrative Tools > Internet Information Services (IIS) Manager.
2. In the Connections pane, expand the computer, right-click Application Pools and select Add Application Pool.
3. Set the Name to `Report Viewer`.
4. Set the .NET Framework version to `.NET Framework v4.0.30319`.
5. Set the Managed pipeline mode to `Integrated`.
6. Select the Start application pool immediately option and click OK.

Parent topic: [Datacap Report Viewer](#)

Single machine installation: Setting up the Datacap Report Viewer website

You must set up the Report Viewer website on Microsoft Internet Information Services (IIS) 7.5.

Procedure

To set up the Report Viewer website:

1. From the Windows Start menu on the Workstation, select Administrative Tools > Internet Information Services (IIS) Manager.
2. In the Connections pane, expand the computer and expand Sites. Then, right-click the Default Web Site and select Add Application.
3. Set the Alias to `Report Viewer`.
4. Click Select and select the Report Viewer Application Pool that you added, then click OK.
5. Set the Physical path by entering or browsing to the Report Viewer installation folder. The default location is `C:\Datacap\RV2`.
6. Click OK to close the Add Application dialog.
7. In the Connections pane, select Application Pools.
8. In the Application Pools pane, select the Report Viewer application pool.
9. In the Actions pane, in the Edit Application Pool section, click Advanced Settings.
10. Ensure that the Microsoft .NET version is set to v4.0.
11. Ensure that Enable 32-Bit Applications is set to True.
12. In the Process Model section, click Browse next to Identity.
13. In the Application Pool Identity window, change the Built-in account to LocalSystem, then click OK.
14. In the Process Model section, set Load User Profile to True.
15. Click OK.
16. In the Connections pane, expand the computer and expand Sites. Then, expand the Default Web Site and select the Report Viewer site, and in the middle pane, double-click Session State.
17. Under Cookie Settings, change the Name to `Report Viewer` or another unique name, then, in the Actions pane, click Apply.
18. In the Connections pane, select the Default Web Site, then, in the Actions pane, under Manage Web Site, click Restart.
19. Confirm that the Web Server, Application Pool, and Default website are started.

Parent topic: [Datacap Report Viewer](#)

Adding the Datacap Report Viewer address as a trusted site

You add the Report Viewer address as a trusted site in a single computer installation of Datacap to prevent Internet Explorer from blocking access to the Datacap website.

Procedure

To add the Report Viewer address as a trusted site:

1. Open Internet Explorer and select Tools > Internet Options.
2. In the Internet Options window, click the Security tab.
3. Select Trusted sites and click Sites.
4. In the Trusted Sites window, enter the default server address (<http://localhost>) in the Add this website to the zone field, then click Add
5. Click Close.
6. Click OK in the Internet Options dialog.

Parent topic: [Datacap Report Viewer](#)

Rulerunner Service (single thread)

The Rulerunner Service is a Datacap component that runs Datacap Studio application background tasks that require no human interaction. Examples of background tasks are VScan, recognition, image pre-processing, validation, and export tasks.

Rulerunner can be configured to run on a single thread with standard Datacap licensing. Configuring Rulerunner to run on multiple threads requires additional licensing.

For demonstration and testing purposes, you can configure Rulerunner on a single machine where all of the Datacap components are installed. The single machine configuration does not allow Rulerunner to take full advantage of the machine's multiple processors. Do not use this configuration as your production configuration.

- [Configure Rulerunner on a single machine](#)
You can configure Rulerunner on a single machine to run Datacap TravelDocs tasks for demonstration and testing purposes.
- [Granting the Rulerunner account the Log On as Service privilege](#)
You must configure the Rulerunner Service properties with the privilege to log on as a local system account. This privilege allows the Rulerunner Service to run as a Windows service.
- [Configuring Rulerunner to run TravelDocs tasks](#)
You can configure Rulerunner to run the TravelDocs PageID task, Profiler task, and Export task.
- [Configuring TravelDocs task profiles](#)
You can ensure that the Datacap Application Manager displays only those task profiles that you want Rulerunner to run.
- [Starting the Rulerunner Service](#)
After you set up or change the Rulerunner Service, go to the Rulerunner Manager and start the Rulerunner service.
- [Monitoring batches during Rulerunner processing](#)
Monitor your batches using the Datacap Web Job Monitor to watch batches change status as they are processed by Rulerunner.

Parent topic: [Installing Datacap on one machine](#)

Configure Rulerunner on a single machine

You can configure Rulerunner on a single machine to run Datacap TravelDocs tasks for demonstration and testing purposes.

You can set up a single Rulerunner thread to run Datacap TravelDocs tasks and to monitor Rulerunner processing when all Datacap components are installed on a single machine. In this configuration, the Rulerunner Service runs using the LocalSystem account.

The process is made up of the following steps:

1. Configure the Rulerunner account.
2. Configure Rulerunner to run Datacap TravelDocs tasks.
3. Create Datacap TravelDocs batches.
4. Start Rulerunner.
5. Monitor Rulerunner processing.

Parent topic: [Rulerunner Service \(single thread\)](#)

Granting the Rulerunner account the Log On as Service privilege

You must configure the Rulerunner Service properties with the privilege to log on as a local system account. This privilege allows the Rulerunner Service to run as a Windows service.

About this task

This procedure provides instructions on how to ensure the Local System account that is used by the Rulerunner Service is granted the Log On as a Service privilege.

Procedure

1. From the Start menu, select Control Panel > Administrative Tools > Services. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.
2. Right-click the Rulerunner Service and select Properties.
3. Click the Log On tab.
4. Select or ensure that the Local System account is selected and click Apply.
5. Click OK to close the Properties dialog.
6. Close the Services window.

Parent topic: [Rulerunner Service \(single thread\)](#)

Configuring Rulerunner to run TravelDocs tasks

You can configure Rulerunner to run the TravelDocs PageID task, Profiler task, and Export task.

Before you begin

Before configuring Rulerunner, [start or ensure the Datacap Server Service is started](#).

About this task

This procedure provides instructions on how to configure Rulerunner to run the TravelDocs PageID, Profiler, and Export tasks on a single thread.

Your license determines the number of threads that you can configure in Rulerunner Manager set up.

Procedure

To configure Rulerunner to run TravelDocs tasks:

1. From the Start menu, select IBM Datacap Services > Datacap Rulerunner Manager.
 - a. If User Account Control (UAC) is enabled, click Yes to close the User Account Control window.
 - b. If the Status is Running, click Stop.
 - c. If the Status is Stopped, continue with the next step.
2. Click the Rulerunner Login tab.
3. Select the Datacap Authentication option to enable the login credential fields.
4. For the Datacap, User ID, Password, and Station ID, enter `admin`, `admin`, and `1`, and click Connect.
5. Click the Workflow:Job:Task tab. The names of the applications from the `datacap.xml` file are displayed in one pane. The other pane does not contain threads the first time that you use Rulerunner Manager.
6. Click the check box next to the TravelDocs application. The application tree expands with the Server, Administrator database, and Engine database selected.
7. Right-click in the pane and select Thread, then select Add Thread.
8. In the other pane, expand the TravelDocs application, click the check box next to the PageID task, Profiler task, and Export task in both the Main Job and the Web Job.
9. Click and drag the Main Job onto the thread.
10. Click Save to save your changes, and then click Yes to save the configuration file.
11. Click the Rulerunner Login tab and click Disconnect.
12. Close the Rulerunner Manager window.

What to do next

From a workstation, start your Datacap client application using your admin user ID, password and station ID. Run your application or applications so that there are batches pending for the tasks that Rulerunner is configured to process.

Parent topic: [Rulerunner Service \(single thread\)](#)

Configuring TravelDocs task profiles

You can ensure that the Datacap Application Manager displays only those task profiles that you want Rulerunner to run.

Procedure

To configure TravelDocs task profiles:

1. From the Start menu, select IBM Datacap Services > Datacap > Datacap Application Manager.
 - a. If User Account Control (UAC) is enabled, click Yes to close the User Account Control window.
2. Select the TravelDocs application. The Main tab displays the paths.
3. Click the Rulerunner tab. This tab displays the task profiles that are processed by Rulerunner.
4. Click the red X next to the VScan profile name, and click Yes to confirm you want to remove the task.
5. Confirm that the PageID task, Profiler task, and Export task are displayed.
6. Close the Datacap Application Manager.

What to do next

[Stop](#) and [restart](#) the Datacap Server Service before starting the Datacap application.

Parent topic: [Rulerunner Service \(single thread\)](#)

Starting the Rulerunner Service

After you set up or change the Rulerunner Service, go to the Rulerunner Manager and start the Rulerunner service.

Procedure

Follow this procedure to start the Rulerunner Service.

1. Go to Start > IBM Datacap Services > Datacap Rulerunner Manager. If the User Account Control window opens, click Yes.
2. If the Status is Stopped, click Start. The Status changes to Running.
3. Close the Rulerunner Manager window.

Parent topic: [Rulerunner Service \(single thread\)](#)

Monitoring batches during Rulerunner processing

Monitor your batches using the Datacap Web Job Monitor to watch batches change status as they are processed by Rulerunner.

Parent topic: [Rulerunner Service \(single thread\)](#)

Installing and configuring in a client/server environment

The typical installation configuration for Datacap is one where the various Datacap software components are installed on dedicated machines. These software components include application servers, web servers, database servers, scan workstations, verify workstations, etc.

The following installation sections provide information on installing Datacap on multiple machines in a distributed environment.

After you install the Datacap software from the installation wizard, you must apply the software license by going to the IBM Datacap Services folder, opening the Datacap License Manager, and selecting your license options. The IBM® Datacap *.swidtag license files are in the C:\Datacap\iso-swid folder.

- [Datacap installation and configuration in a client/server environment](#)
The list of Datacap components you can install in a client/server environment includes the Datacap Web Client, Datacap Application Manager, and Rulerunner. You can also install Maintenance Manager, Report Viewer, and Datacap Web Services.
- [Datacap Web Services installation steps](#)
The Datacap Web Services installation in a client/server environment requires you to follow several steps, including preparation, installation, configuration, and verification steps.
- [Client/server installation checklist](#)
This checklist provides a summary of the account and configuration settings that are required to run Datacap software components in a client/server environment for demonstration, proof of concept, development and test purposes.
- [Datacap installation command-line parameters](#)
You can run the Datacap installation program Setup.exe file by using command-line parameters from the command line.

Parent topic: [Installing](#)

Datacap installation and configuration in a client/server environment

The list of Datacap components you can install in a client/server environment includes the Datacap Web Client, Datacap Application Manager, and Rulerunner. You can also install Maintenance Manager, Report Viewer, and Datacap Web Services.

Datacap can be installed, configured, and run in a client/server environment. Datacap includes many components, some of which are required for Datacap to function. Some components are not required for Datacap to function but can be installed to meet your processing requirements. You can install and configure some or all of these components for demonstration, proof of concept, development, and test purposes.

- Datacap Server
- Datacap Clients, including Datacap FastDoc, Datacap Studio, and Datacap Maintenance Manager.
- Datacap Rulerunner Server
- Datacap Web Server
- Datacap Web Service
- Datacap Windows Service
- Report Viewer
- Connectors

You can set up and run the separately licensed Datacap applications, including the Datacap Accounts Payable application.

For a checklist of the account and configuration settings, see the [Client/server installation checklist](#).

After you install the Datacap software from the installation wizard, you must apply the software license by going to the IBM Datacap Services folder, opening the Datacap License Manager, and selecting your license options. The IBM® Datacap *.swidtag license files are in the C:\Datacap\iso-swid folder.

- [Datacap client/server configuration](#)
Datacap can be configured in many ways, such as installing components on separate machines, installing multiple components on a single machine, or installing a single component on multiple machines.
- [Installing and configuring Datacap on a client and server](#)
The steps to install and configure Datacap include installing components such as the Datacap Web Client, Report Viewer, Maintenance Manager, and Rulerunner. The sample TravelDocs application that is delivered with Datacap provides you with an easy way to confirm that you set up Datacap correctly.
- [Installation instructions for Datacap server](#)
Installation instructions for the Datacap server include ensuring that an account with administrator privileges is granted Full Control access to all of the machines on which the Datacap components are being installed.
- [Client/server environment: Datacap Web Client installation and configuration](#)
You can run tasks from a computer on which the Internet Explorer browser is installed when you set up the Datacap Web Client. In a single machine configuration that is used for testing and demonstration purposes, Datacap Web Client runs using the LocalSystem account.
- [Installing and configuring Datacap Navigator](#)
Datacap Navigator is a web client for Datacap based on IBM Content Navigator. You set up the Datacap Navigator client by loading a plug-in and configuring repositories and desktops in IBM Content Navigator.
- [Installing the developer workstation software components](#)
Run the Datacap installation program wizard on the workstation of a developer to install the Datacap software components.

- [Copying the application to the Datacap Server](#)
The following information provides instructions for copying specific applications. However, the instructions are equally applicable to a customized application, and you can refer to any one of the topics for copying your custom application. You need to start Datacap Studio, start the Datacap Studio Application Copy wizard, and copy the application from the developer workstation to the server. The application can then be accessed by other workstations that are running the Datacap Client or Datacap Web Client, and by other services and processes.
- [Complete the Datacap server configuration](#)
You must update the datacap.xml file on the Datacap server to add references to applications. You must also set up security on the folder that contains the applications.
- [Complete the Datacap Web Client server configuration](#)
To complete the Datacap Web Client server configuration, you must set the location of the Datacap.xml file. Then, you must restart Internet Information Services.
- [Configuring and testing the remote workstation](#)
You can configure and test a remote workstation that uses Internet Explorer to access Datacap Web Client.
- [Installing the Datacap client on the user workstation](#)
You can run the Datacap installation program wizard on a user workstation to install the necessary Datacap client software.
- [Installing and configuring Datacap Report Viewer](#)
You can install and configure Report Viewer to run locally or in a Datacap client/server environment so you can use the browser-based Datacap Web Client for processing.
- [Installing and configuring the Rulerunner Service](#)
The Rulerunner Service is a Datacap component that runs Datacap Studio application tasks in the background.
- [Installing and configuring the Datacap Fingerprint Service](#)
The Datacap Fingerprint Service, with actions in the Autodoc actions library, supports the ability to cache and use the fingerprints from one or multiple Datacap applications simultaneously. The Fingerprint Service eliminates the fingerprint load time for all but the first batch that is processed by an application.
- [Installing and configuring the Datacap Web Client upload service](#)
After you scan files and create batches with Datacap, you can upload the files by using the Datacap Web Client upload service.
- [Starting Datacap Studio](#)
This procedure provides instructions on how to start the Datacap Studio software component.
- [Installing and configuring Datacap Maintenance Manager](#)
Datacap Maintenance Manager is a Datacap software component that sets up batch monitoring, status notification, and automatic deletion of completed batches.
- [Configuring authentication for Datacap](#)
You can configure Datacap to use Datacap authentication or to use one of the external authentication methods.
- [Installing and configuring IBM Datacap Advanced Handwriting Recognition 9.0.1](#)
The IBM Datacap Advanced Handwriting Recognition 9.0.1 is an add-on for handwriting recognition for checks.

Parent topic: [Installing and configuring in a client/server environment](#)

Related concepts:

[Client/server installation checklist](#)

Datacap client/server configuration

Datacap can be configured in many ways, such as installing components on separate machines, installing multiple components on a single machine, or installing a single component on multiple machines.

The multiple machine configuration that is used throughout the client/server examples that are provided in these tasks includes the following machines:

- The Datacap Server that hosts the main application settings file (datacap.xml), the applications, batch files, and databases.
- The Datacap Web Server and the Report Viewer.
- The developer workstation that hosts the Datacap clients, including Datacap Studio, Datacap Desktop, FastDoc, and Maintenance Manager.
- Workstations that host Datacap clients, such as Datacap Desktop, your browser, and the Datacap Web Client Upload Service.
- The Rulerunner server hosts the Rulerunner Service, which runs tasks that do not require human interaction.
- The Fingerprint Service server that hosts the Fingerprint Service.

You must also create accounts and permissions that are needed for the following types of users, processes, and services.

- Implementation and development personnel with administrative rights on all machines
- Users
- Datacap Server service
- Datacap Web Client
- Rulerunner Service
- Maintenance Manager
- Report Viewer
- Fingerprint Service
- Datacap Web Services

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Installing and configuring Datacap on a client and server

The steps to install and configure Datacap include installing components such as the Datacap Web Client, Report Viewer, Maintenance Manager, and Rulerunner. The sample TravelDocs application that is delivered with Datacap provides you with an easy way to confirm that you set up Datacap correctly.

Procedure

To install and configure Datacap on a client and server:

1. Ensure that you meet the prerequisites that are detailed in *Installation and Configuration Prerequisites*.
2. Install the Datacap Server software component on the server.
3. Configure Datacap on the server.
4. Install Datacap Web Client on the web server.
5. Configure Datacap Web Client on the web server.
6. Install the Datacap client software component, the separately licensed applications, and the connectors for which you have licenses on the developer workstation.
Important: Include the Datacap Connector for eMail and Electronic Documents connector, which is required for Datacap Studio.
7. Copy the TravelDocs application from the developer workstation to the server.
8. Complete the Datacap Web Client server configuration.
9. Complete the configuration.
10. Run the TravelDocs application to validate the server installation.
11. Configure Internet Explorer on the developer workstation.
12. Run Datacap Web Client TravelDocs to validate the web server installation.

13. Install and configure the Datacap client software component on a user's workstation.
14. Install Report Viewer on the web server.
15. Configure Report Viewer on the web server.
16. View Report Viewer reports to validate the Report Viewer installation.
17. Install Rulerunner on the Rulerunner server.
18. Configure Rulerunner to run TravelDocs both single thread and multi-thread tasks.
19. Run TravelDocs tasks to validate the Rulerunner installation.
20. Configure Rulerunner to run your application tasks.
21. Install Maintenance Manager on the developer workstation.
22. Set up Maintenance Manager application or actions on the developer workstation.
23. Run Maintenance Manager actions to validate the installation.
24. Configure Datacap to use Windows Authentication.
25. Use Fingerprint Maintenance Tool.
26. Install, configure, and test the Fingerprint Service on the Fingerprint Service server.
27. Update an application to use the Fingerprint Service.
28. Configure and distribute the Datacap Web Client Upload Service.
29. Install, configure, and test Datacap Web Services.
30. Install, configure, and test FastDoc.
31. Install, configure, and run the Datacap Accounts Payable application.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Related concepts:

[Installation and configuration prerequisites](#)

Installation instructions for Datacap server

Installation instructions for the Datacap server include ensuring that an account with administrator privileges is granted Full Control access to all of the machines on which the Datacap components are being installed.

Before you install Datacap server, ensure that you meet the prerequisites that are detailed in *Installation and Configuration Prerequisites*.

- [Ensure an account exists for the Datacap server service](#)
When you perform the default Datacap installation and configuration, the Datacap server service uses the Local System account to log onto the server.
- [Installing Datacap server](#)
Run the Datacap installation program wizard on a server to install the necessary Datacap software components.
- [Configuring Datacap on the server](#)
To configure Datacap on the server, you must grant the appropriate permissions to the Datacap server service and grant the appropriate permissions to folders and users.
- [Configuring a Datacap application for Datacap Mobile](#)
This topic provides information specific to configuring the parts of your Datacap application to work with Datacap Mobile. Good knowledge of Datacap application configuration tasks and Datacap Studio or FastDoc are expected. Also, Datacap Mobile SDK documentation is included in both the iOS and Android SDK archive files that can be found in the Datacap installation tree under the \Datacap\MobileSDK\ folder, and for the latest release, on [IBM Fix Central](#).
- [Exporting encryption keys](#)
You must generate security keys that allow Datacap to encrypt and decrypt passwords. To replace existing keys with new keys, you can specify a parameter to generate encryption keys to a local store. You can specify a different parameter to export the encryption keys from the local keystore to a file that can be imported to other computers.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Related concepts:

[Installation and configuration prerequisites](#)

Ensure an account exists for the Datacap server service

When you perform the default Datacap installation and configuration, the Datacap server service uses the Local System account to log onto the server.

In a client/server environment, create or ensure that a domain/Windows account exists for the Datacap server service. Datacap does not require that a unique Windows account be set up for the Datacap server service. The Datacap server service can use any Windows account provided that account can be set up with the appropriate sharing and security permissions. If you have multiple Datacap servers, you can set up individual Windows accounts or you can set up a single Windows account that is shared.

Parent topic: [Installation instructions for Datacap server](#)

Installing Datacap server

Run the Datacap installation program wizard on a server to install the necessary Datacap software components.

About this task

The components that you install are the Datacap server component and the separately licensed connectors for which you have a license. If you have multiple Datacap servers, repeat these instructions on each server.

Procedure

1. Make the installation package available on your network, or insert the Datacap CD in the server's CD/DVD drive. If the installation process does not start automatically, or if the package is on the network, open Windows Explorer, navigate to and double-click the Setup.exe. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.
2. Select the appropriate language. Then, click OK. The language that you select determines the language that is displayed by the installation program wizard during the installation process.
3. When additional, redistributable software is required, the installation program wizard displays a list of the items to be installed. Click Install.
4. Click Next.
5. Accept the license agreement and click Next.
6. Select the Custom option. Then, click Next.
7. Exclude all components from the installation process except Datacap Server and the separately licensed connectors for which you have a license.
8. Click Next.
9. Click Install.
10. Click Finish.

Parent topic: [Installation instructions for Datacap server](#)

Configuring Datacap on the server

To configure Datacap on the server, you must grant the appropriate permissions to the Datacap server service and grant the appropriate permissions to folders and users.

- [Granting Datacap server service the Log On as a Service privilege](#)
You must configure the Datacap Server properties to ensure the domain account that is used by the Datacap server service is granted the Log On as a Service privilege.
- [Setting up sharing permissions on the Datacap folder](#)
You must set up sharing permissions for the C:\Datacap folder by setting sharing permissions Datacap Properties Sharing window
- [Setting up security on the Datacap folder](#)
You must set up security for the shared C:\Datacap folder by setting security permissions in the Datacap Properties Securities window.
- [Setting up security on the Datacap\RRS folder](#)
You must set up the appropriate security permissions for the C:\Datacap\RRS folder on the server when the operating system is Windows 2008.

Parent topic: [Installation instructions for Datacap server](#)

Granting Datacap server service the Log On as a Service privilege

You must configure the Datacap Server properties to ensure the domain account that is used by the Datacap server service is granted the Log On as a Service privilege.

About this task

If you have multiple Datacap servers, repeat this process for each server.

Procedure

To grant Datacap server service the Log On as a Service privilege:

1. From the Start menu on the server, click Control Panel > Administrative Tools > and double-click Services.
2. If you want the Datacap server to start automatically when the server is restarted, select Datacap Server and change the Startup Type to Automatic and save your change.
3. Right-click Datacap Server and select Properties.
4. Click the Log On tab, then select This account.
5. Locate or enter the domain name, user name, and password of the Windows account to be used by the Datacap server service, and click Apply.

Parent topic: [Configuring Datacap on the server](#)

Setting up sharing permissions on the Datacap folder

You must set up sharing permissions for the C:\Datacap folder by setting sharing permissions Datacap Properties Sharing window

Procedure

1. On the server, start Windows Explorer, navigate to, and right-click the C:\Datacap folder and select Properties.
2. Click the Sharing tab. Then, click Advanced Sharing. When User Account Control (UAC) is on, the User Account Control window is displayed. Click Yes.
3. Click Share this Folder and keep `Datacap` as the Share name.

4. Click Permissions. Ensure that the NETWORK SERVICE and local IUSR accounts are set to allow Full Control.
5. Ensure that the domain/Windows user IDs of developers are set to allow Full Control.
6. Ensure that the domain/Windows user ID of Datacap server service is set to allow Full Control.
7. Ensure that the domain/Windows user ID of Datacap Web Client is set to allow Read.
8. When the batches folders for the application are staying on server in C:\Datacap\Application path, ensure that the domain/Windows user IDs of Datacap users are set to allow Full Control.

Parent topic: [Configuring Datacap on the server](#)

Setting up security on the Datacap folder

You must set up security for the shared C:\Datacap folder by setting security permissions in the Datacap Properties Security window.

Procedure

1. On the server, start Windows Explorer, navigate to, and right-click the C:\Datacap folder and select Properties.
2. Click the Security tab. Then, click Edit. When User Account Control (UAC) is on, the User Account Control window is displayed. Click Yes.
3. Ensure that the domain/Windows user IDs of developers who can change existing Datacap applications are set to allow Read & Execute. This developer cannot create new applications.
4. Ensure that the domain/Windows user IDs of developers who can create new Datacap applications in the C:\Datacap folder are set to allow Full Control.
5. Ensure that the domain/Windows user ID of Datacap server service is set to allow Read & Execute.
6. Ensure that the domain/Windows user ID of Datacap Web Client is set to allow Read & Execute.
7. If the batches folders stay on the server in the C:\Datacap\Application path, ensure that the domain/Windows user IDs of Datacap users are set to allow Read & Execute.

Parent topic: [Configuring Datacap on the server](#)

Setting up security on the Datacap\RRS folder

You must set up the appropriate security permissions for the C:\Datacap\RRS folder on the server when the operating system is Windows 2008.

Procedure

1. On the server, start Windows Explorer, navigate to and right-click the C:\Datacap\RRS folder and select Properties.
2. Click the Security tab to display it. Then, click Edit. When User Account Control (UAC) is on, the User Account Control window is displayed. Click Yes.
3. Add or ensure that the NETWORK SERVICE and local IUSR accounts are set to allow Read & Execute.

Parent topic: [Configuring Datacap on the server](#)

Configuring a Datacap application for Datacap Mobile

This topic provides information specific to configuring the parts of your Datacap application to work with Datacap Mobile. Good knowledge of Datacap application configuration tasks and Datacap Studio or FastDoc are expected. Also, Datacap Mobile SDK documentation is included in both the iOS and Android SDK archive files

that can be found in the Datacap installation tree under the \Datacap\MobileSDK\ folder, and for the latest release, on [IBM Fix Central](#).

- [Configuring property display for mobile](#)
This section provides information about how to configure property display for mobile.
- [Configuring bar code based classification](#)
Using Datacap Studio or FastDoc, you can associate a specific bar code type and value to a page type for Datacap Mobile to automatically read the bar code and assign the matching page type at snap or import time. When you open a claim with your auto-insurance company, they assign a unique claim number to your claim and reference it in a bar code on their claim form every time they exchange correspondence with you, such as when requesting additional supporting documents from you. This is to help them sort out the returning mail and automatically route it to your claim agent. To support that case in your Datacap application, you need to associate a regular expression matching the numbering pattern of the claim to the claim form page type (for instance). Datacap Mobile then reads the barcode value, executes the regular expression against it, and if it finds a match automatically assigns the claim form page type to the image at snap time.
- [Configuring template based zonal recognition](#)
Template based zonal recognition is used to automate the extraction of machine-printed characters from fixed forms. Using Datacap Studio or FastDoc you import an image of the type of form you want, captured on your mobile device. Then you need to create zones and associate them to the data fields of the page type associated to the image. This constitutes a "zonal template". For example, you can define a zonal template for a "Claim form" page type to capture the claimant's first and last names, policy number, claim number from a bar code, car model, make, and license plate number appearing on the claim form. Datacap Mobile reads the zone definitions associated with the page type, at snap or import time, or every time a new page type is selected in the Folder screen (Document Assembly screen), and automatically performs character recognition associated to each zone.
- [Configuring recognition of identification documents](#)
Datacap Mobile can automatically read the information encoded in identification documents and extract data.
- [Configuring check recognition](#)
Datacap Mobile extracts key data from checks, such as the date, courtesy and legal amounts, payee name and the routing, account, and check numbers which are components of the Magnetic Ink Character Recognition (MICR) line at the bottom of checks.
- [Configuring the geolocation](#)
Datacap Mobile transfers the current location of the device to the Datacap application and images.

Parent topic: [Installation instructions for Datacap server](#)

Basic principles

Datacap Mobile works off the Setup DCO

Datacap Mobile offers a lot of flexibility and configures itself as per the document model defined in the Datacap (server) application that it connects to. Using Datacap Studio or FastDoc you can, as an administrator, configure Datacap to address the specific use case and nature of the business documents to be processed.

For example, for an account opening application, you can define the document types and data that need to be captured from the account application form, an identification card, a pay stubs, etc. required to open the account.

This document and data model is called the application "Document Hierarchy" or "Setup DCO", in Datacap. When connecting the Datacap Mobile app to the Datacap application on the Datacap Server, you are now able to use the document, page, and field types that have been defined in the app's pick lists.

In addition to the Document Hierarchy, you can configure processing capabilities unique to your application. For example, and more specifically for mobile, you can configure a bar code to automatically identify your account application form and associate to the form a zonal recognition template to automatically capture all the data you need from the form. You can do the same to automate the recognition of a US Driver License, for example.

Auto-classification in Datacap Mobile

Datacap Mobile automatically assigns a type to a document based on recognizing a bar code value or on the defaults defined on the Datacap Document Hierarchy.

- If barcode-based classification is not set up in the Datacap application, then Datacap Mobile by default assigns the first document type and page type it finds in the Document Hierarchy sequence. If the user manually specifies a document type in the Camera screen, then Datacap Mobile by default assigns the first page type within that document type.
- If bar code based classification is set up, Datacap Mobile determines the page type first based on the detected bar code value which, in turn, causes the parent document type to be assigned. For example, in an account opening application, the critical pages for classification are the application form and the back side of the driver license, because they each have a uniquely identifiable bar code. When capturing the driver license's back side first, Datacap Mobile assigns the "DL Back" page type and automatically switches the document type to "Driver License" in the Camera screen. It remains the default setting until another type of page is detected and, possibly, the document is switched. The driver license's front side is not critical for classification; it will be assigned the current default page type, and hence, document type.

Character recognition and working with mobile images

To configure the mobile input channel of your Datacap application, and work with image fingerprints in Datacap Studio or FastDoc, you need to get images that originate from a mobile device, especially when configuring zonal OCR. Using image fingerprints scanned from a scanner might work, but mostly won't.

This is because OCR engines operate best on images with a resolution in the 200-300 DPI range. They have been designed so that within this range they can fairly accurately segment the images to determine lines of text, character attributes, such as font types and sizes, or to discriminate between small size text and noise, specks, etc. mostly based on measuring the size of what they are looking at. So, it is essential to be able to relate these attributes to the physical dimensions of the documents, to normalize them to a fixed DPI value, typically 300.

Unfortunately, contrary to a scanner that can precisely detect the physical dimensions of the document placed on its platen and acquire an image at a specific DPI value, a mobile device's camera can snap a document from any distance and therefore is not able to assign an accurate resolution to the images it produces. For a given camera pixel width and height, the farther and smaller the document appears in the camera screen, the lower the resolution gets after deskewing and cropping.

To deal with this problem, Datacap Mobile, in automatic mode, only snaps images that occupy about 70% of the screen surface area, ensuring that "enough pixels" of the document are acquired to match an assumed resolution of 200-300 DPI, and the skewing of the document remains within acceptable limits. This ensures that a perfect quadrangle can be reconstructed and well-proportioned against the original image.

Also, for some use cases it makes sense to complement recognition on the mobile side, say, extraction of a few fields to present for repair to the user, with recognition of additional fields and validation/verification on the server side. In this situation, it is important to normalize the images to 300 DPI to get to a common reference for accurately positioning the zones on the server side and in the other Datacap clients that will be used for verification, such as Datacap Desktop or Navigator. This is especially important for small-size documents, such

as driver licenses, that can be shot in landscape mode to occupy the whole camera display. In this case the apparent DPI is much higher than normal and confuses the OCR engines used in Datacap server.

For this purpose, it is advisable to insert the `SetImageDPIByWidth` action from the `ImageConvert` library to resize proportionally the incoming images to 300 DPI based on the known physical dimensions of the document. Assigning fixed physical dimensions is not an issue when dealing with forms as they typically have a fixed and predictable layout.

Getting started quickly with the mobile template

The Mobile Template is a very simple Datacap application that allows you to get started with configuring a Datacap application for Datacap mobile very quickly. It includes a basic one-document/one-page document hierarchy that you can expand from, as well as jobs, task profiles, and rulesets that have been preset for that purpose.

Jobs are the entry points in a Datacap application, and there can only be one specified per Application you define in Datacap Mobile. These are the jobs defined in the Mobile Template:

- **Mobile** only is what you want to use when you do all the processing, that is, snap, deskew, classify, OCR, assemble, repair, and upload, on the mobile side, and only need to verify and export to a repository on the server side. Once the images are uploaded, together with extracted field/property data on the Datacap server, it essentially rescales them with the `SetImageDPIByWidth` action and sends them to Verify without any other processing. This is very fast. You can even remove the Verify task to have a direct process to the repository.
Note: Even though the Identify Page compiled ruleset does not run on the server side (that is, not included in this job), you still use it to set up automatic bar code recognition on the mobile side. Datacap Mobile only needs to read from the Setup DCO the bar code processing variables written by the Identify Pages to the batch level (selected bar code types) and to each mapped page type (the regular expression to extract the bar code value)
- **Mobile and server Color** is what you would want to use for a combination of mobile and server-side processing. For example, if you want to process content-intensive document, with no predictable layout, you would likely need to recognize the full-page content of mobile images and process it server-side. In addition to the processing described above, this job allows to run the full power of the Identify Pages ruleset (blank page, fingerprints, keywords, content classification, etc.), server-side full-page OCR with structural analysis, extraction of content by location, document assembly, and merging of server-side and mobile-side extracted fields, and validation rules.
- **Mobile and server Bitonal** is similar to the above, but adds image enhancement and conversion to black and white images, required in certain cases
- **Import Color** is used for testing, to import preexisting, deskewed color mobile images from your local file system in Datacap. This is useful for testing server-side processing without having to recapture images all the way from mobile every time. You just need a good set and re-import it to test as you refine the server-side processing.
- **Import Bitonal**- same as above but for black and white images

You can download the Mobile Template from developerWorks and drop it in the [...]Datacap/Templates folder of the Datacap folder structure. After adding a reference to it in the `datacap.xml` file, it automatically displays in the list of templates that you can choose from when you use the Datacap Application Wizard to create your application.

Configuring property display for mobile

This section provides information about how to configure property display for mobile.

Hiding fields on mobile only

In your Datacap application, you can set the *STATUS* variable of any object of the document hierarchy to "-2" to hide it from mobile users only. This can be useful when you build applications with recognition and indexing on both the mobile and server sides and you do not want server-side fields that are used by internal users to be displayed to mobile users.

Underscore character "_" converted to blank space

If you want to enhance the document, page, and field symbolic names of the Setup DCO without going to the extent of using labels, Datacap converts underscores to blank spaces for you.

Assigning friendly names to fields/properties

Datacap Mobile supports displaying friendly names by using the *label* variable that is used in Datacap applications. For example, in Check Processing, to grab the check account number from the "MICRData1" component of the MICR line, add a variable named *label*, using Datacap Studio, and assign the value of "Account Number" for it to be displayed in Datacap Mobile.

Parent topic: [Configuring a Datacap application for Datacap Mobile](#)

Configuring bar code based classification

Using Datacap Studio or FastDoc, you can associate a specific bar code type and value to a page type for Datacap Mobile to automatically read the bar code and assign the matching page type at snap or import time. When you open a claim with your auto-insurance company, they assign a unique claim number to your claim and reference it in a bar code on their claim form every time they exchange correspondence with you, such as when requesting additional supporting documents from you. This is to help them sort out the returning mail and automatically route it to your claim agent. To support that case in your Datacap application, you need to associate a regular expression matching the numbering pattern of the claim to the claim form page type (for instance). Datacap Mobile then reads the barcode value, executes the regular expression against it, and if it finds a match automatically assigns the claim form page type to the image at snap time.

Procedure

To configure bar code classification, complete the following steps:

1. Open Datacap Studio and then log in to your application.
2. Allow Global Rules to load.
3. Right-click Identify Pages and choose Install in application.
4. Scroll down to Identify Pages. Right-click Identify Pages and choose Settings. The bar code configuration displays.
5. Expand Barcode Recognition.
6. Select Types and the bar code type.
7. Select Orientation and the type of orientation.
8. Check the Regular Expressions check box.
9. In Mappings, enter a regular expression in the Barcode Value field. For example, to identify all claims with a number looking like "CL-4326543", with "CL-" always fixed, followed by a number, enter a pattern of the form "CL-|d*", which matches a case sensitive string starting with "CL-" followed by any number of digits. If you want to match a pattern of exactly 7 digits after "CL-", you can use "CL-|d{7}".
10. Select Page Type.
11. Click Save.

Bar codes supported in Datacap Mobile

The following bar codes are supported in Datacap Mobile:

AZTEC, CODABAR, CODE128, CODE39, CODE93, DATAMATRIX, EAN-8 EAN-13, INTERLEAVED 2 OF 5 (ITF), PDF417, QR CODE, UPC-A, and UPC-E.

Parent topic: [Configuring a Datacap application for Datacap Mobile](#)

Configuring template based zonal recognition

Template based zonal recognition is used to automate the extraction of machine-printed characters from fixed forms. Using Datacap Studio or FastDoc you import an image of the type of form you want, captured on your mobile device. Then you need to create zones and associate them to the data fields of the page type associated to the image. This constitutes a "zonal template". For example, you can define a zonal template for a "Claim form" page type to capture the claimant's first and last names, policy number, claim number from a bar code, car model, make, and license plate number appearing on the claim form. Datacap Mobile reads the zone definitions associated with the page type, at snap or import time, or every time a new page type is selected in the Folder screen (Document Assembly screen), and automatically performs character recognition associated to each zone.

Procedure

You need an image sample captured and deskewed on your phone for each page type on which you want to associate an OCR zonal template.

To configure predefined field templates, complete the following steps :

1. Open Datacap Studio and then log in to your application.
2. Click the Zones tab to open the Fingerprints view.
3. Add a mobile fingerprint (or select an existing one).
4. Right-click a top level fingerprint class and choose Add fingerprint.
5. Browse to your sample image and open it.
6. Click the padlock icon to lock the document hierarchy.
7. Expand the hierarchy to display fields.
8. Select a field and draw a rectangular zone inside the Image View tab.
9. Repeat the step above for each field on which you want to perform OCR.
10. Save the document hierarchy.
11. Click the padlock icon to unlock the document hierarchy.

Important: For a zonal template to function properly, it is essential to position the extraction zones on the fingerprint accurately. Ensure that the incoming images are proportioned same as the fingerprint, so that Datacap Mobile can grab the image snippets for OCR in the proper locations after deskewing. Note that Datacap Mobile gets positional information from the *Default_Position* variable for each field in the Setup DCO. This variable gets automatically updated with new position information by Datacap Studio or FastDoc each time the zone position is adjusted in the fingerprint. Be aware that if you use a single fingerprint (zonal template) for both mobile and standard scanning, changes applied last get written to the *Default_Position* variable and might not be appropriate for mobile and vice-a-verse.

Helping the OCR engine recognize an arbitrary character sequence

In cases when the content of a field can be a random sequence of characters, it can be difficult for the Datacap Mobile OCR engine to correctly recognize certain characters, such as "1" vs. "I" or "l", "O" (alpha) vs. "0" (numeric), etc., which are not supposed to occur in the dictionaries that have been used

to train the engine. To help the engine handle exceptions and correctly recognize arbitrary sequences of characters, a regular expression that defines the expected pattern can be associated to a Datacap field.

To do so in Datacap Studio, simply add a variable named *hr_Regex* to the field and assign a regular expression as a value. For example for a field that contains a Vehicle Identification Number or VIN, enter:

```
/^(?<wmi>[A-HJ-NPR-Z\d]{3})(?<vds>[A-HJ-NPR-Z\d]{5})(?<check>[\dX])(?<vis>(?  
<year>[A-HJ-NPR-Z\d])(?<plant>[A-HJ-NPR-Z\d])(?<seq>[A-HJ-NPR-Z\d]{6}))$
```

For example, this regular expression will match VINs of the form:

- o 3FADP4AJ3CM159727 (specific to Ford)
- o JHMGE8H39CC039314 (specific to Honda)

Parent topic: [Configuring a Datacap application for Datacap Mobile](#)

Configuring recognition of identification documents

Datacap Mobile can automatically read the information encoded in identification documents and extract data.

Parent topic: [Configuring a Datacap application for Datacap Mobile](#)

Passports

Datacap Mobile can automatically recognize biometric passports by reading the information encoded in the lines of the Machine Readable Zone (MRZ) at the bottom of the photo page of passports. It can extract key data, such as names, passport number, nationality, and expiration date directly from the MRZ. Using Datacap Studio, you just need to define document, page and fields with predefined variables, and when the page type is selected by the user or assigned by default, the recognition engine reads the MRZ lines, populates the fields and checks confidence.

Procedure

To configure passport recognition in Datacap Studio, follow these steps:

1. Add a document type with any symbolic and display names.
2. Add a page type with any symbolic and display names for the ID.
3. Define the following variables for the page above:
 - o *id_type* = *MRZ2*
 - o *id_subtype* = *passport*
 - o *id_nation* = *USA*
 - o *id_page_type* = *code_page*
4. Add the field types of interest with any symbolic and display names for the ID.
5. For each field type of interest, define the following variable (plus optionally a label, as needed): *ID_Key* = *<DL_field>*; Where *<DL_field>* equals any of the following:
 - o *ID_Number*
 - o *ID_First_Name*
 - o *ID_Surname*
 - o *ID_Birthdate*
 - o *ID_Expiration*
 - o *ID_Sex*
 - o *ID_Expedition_Country_Code*

US Driver Licenses

Datacap Mobile can automatically recognize AAMVA-compliant US Driver Licenses by reading the information encoded in the PDF417 bar code printed in the back of the license. It can extract data such as the driver's first name, last name, license number, date of birth, and address, etc. A number of US States still do not comply, or do not fully comply, with the AAMVA standard. Similarly older valid driver licenses compliant with an earlier version of the standard (prior to the 2013 revision), can't be recognized. However, newly issued driver licenses are more and more compliant with the standard or to an extent that is required by Datacap Mobile. The Territories and States that do not comply at all with the standard are: American Samoa, Minnesota, Missouri, and Washington.

Procedure

Using Datacap Studio, you just need to define document, page and fields with predefined variables, and when the page type is selected by the user or assigned by default or through bar code based classification, the recognition engine reads the 2D bar code (of type PDF417), populate the fields, and checks confidence. Follow these steps to configure passport recognition in Datacap Studio:

1. Add a document type with any symbolic or display name.
2. Add a page type with any symbolic name (ex: Driver_License_Back) or display name (ex: Back) for the back (bar code) side of the US Driver License.
3. Define the following variables for the page above:
 - o *id_type = DrivingLicense*
 - o *id_nation = USA*
 - o *id_page_type = code_page*
4. Add the field types of interest with any symbolic and display names for the ID. For each field type of interest, define the following variable (plus optionally a label, as needed): *ID_Key = <DL_field>*; Where *<DL_field>* equals any of the following:
 - o *ID_Number*
 - o *ID_First_Name*
 - o *ID_Surname*
 - o *ID_Birthdate*
 - o *ID_Expiration*
 - o *ID_Sex*
 - o *aamva_field_<data_element>*

Where *<data_element>* can be any additional mandatory or optional data elements that are specified in the AAMVA DL/ID Card Design Standard. Mandatory data elements are:

Data element	Description
DCA	Jurisdiction-specific vehicle class
DCB	Jurisdiction-specific restriction codes
DCD	Jurisdiction-specific endorsement codes
DBA	Document Expiration Date
DCS	Customer Family Name
DAC	Customer First Name
DAD	Customer Middle Name(s)
DBD	Document Issue Date
DBB	Date of Birth
DBC	Physical Description – Sex

DAY	Physical Description – Eye Color
DAU	Physical Description – Height
DAG	Address – Street 1
DAI	Address – City
DAJ	Address – Jurisdiction Code (State)
DAK	Address – Postal Code
DAQ	Customer ID Number
DCF	Document Discriminator
DCG	Country Identification
DDE	Family name truncation
DDF	First name truncation
DDG	Middle name truncation

Here is an example:

Field symbolic name	ID key variable
ID_Number	ID_Key=ID_Number
ID_First_Name	ID_Key=ID_First_Name
ID_Surname	ID_Key=ID_Surname
ID_Birthdate	ID_Key=ID_Birthdate
ID_Expiration	ID_Key=ID_Expiration
ID_Sex	ID_Key=ID_Sex
ID_Eye	ID_Key= aamva_field_DAY
ID_Street_Address	ID_Key= aamva_field_DAG
ID_City_Address	ID_Key= aamva_field_DAI
ID_State_Address	ID_Key=aamva_field_DAJ
ID_Zip_Address	ID_Key=aamva_field_DAK

French National ID Card recognition (iOS only)

Datacap Mobile can automatically recognize French National ID Cards (Carte Nationale d'identité), by reading the MRZ lines at the bottom of the photo side. It can extract key data, such as first name, last name, ID card number, and birth date, directly from the MRZ. Using Datacap Studio, you need to define document, page and fields with predefined variables. When the page type is selected by the user or assigned by default, the recognition engine reads the MRZ lines, populates the fields and checks confidence.

Procedure

Follow these steps to configure passport recognition in Datacap Studio:

1. Add a document type with any symbolic and display names.
2. Add a page type with any symbolic and display names for the ID.
3. Define the following variables for the page above:
 - o *id_type = PersonalId*

- *id_subtype = id_card*
 - *id_nation = FRA*
 - *id_page_type = code_page*
4. Add the field types of interest with any symbolic and display names for the ID.
 5. For each field type of interest, define the following variable (plus optionally a label, as needed): *ID_Key = <DL_field>*; Where *<DL_field>* equals any of the following:
 - *ID_Number*
 - *ID_First_Name*
 - *ID_Surname*
 - *ID_Birthdate*
 - *ID_Expiration*

Configuring check recognition

Datacap Mobile extracts key data from checks, such as the date, courtesy and legal amounts, payee name and the routing, account, and check numbers which are components of the Magnetic Ink Character Recognition (MICR) line at the bottom of checks.

Parent topic: [Configuring a Datacap application for Datacap Mobile](#)

Understanding the check processing

The capability to process checks on Datacap Mobile requires a connection to the Datacap server because the recognition process is executed on the server side, via an optional, add-on component called IBM Datacap Advanced Handwriting Recognition. Hence, make sure that it is installed on the Datacap server.

After Datacap Mobile has captured the image of a check, it makes a "transactional" call to the Datacap server to pass the image and receive the extracted data back for validation by the user. Checks are typically small in size and this round trip only takes a few seconds. To make the interface between DatacapMobile and any type of server-side processing easy and flexible, Datacap has a special helper application named 'Transaction', which installs directly on the Datacap server and acts as a front end to the Check Processing functions.

Transaction application

Its role is to:

1. receive the check image and some check-specific variables, such as the *width* and *DPI* and check country, passed on untouched by Datacap Mobile from the Datacap application it is connected to (remember, the one you log on to in Applications)
2. perform some check-specific image pre-processing functions, such as resizing and setting a DPI value
3. invoke the check recognition action and get the results
4. send back the extracted data (only; the image is discarded) once the check has been recognized to Datacap Mobile.
5. This application does not require any special configuration.

Configuring your application

Follow these steps in Datacap Studio for configuring the Datacap application for check processing.:

1. Add a page type with any symbolic name, for example, Check
2. Add a variable to that page type named *ProcessChecks* with value set to True. This instructs Datacap Mobile to process the document as a check, that is, to send the image to the Datacap server for processing and get the results back for display and repair.

3. Add a variable to that page type named *hr_CheckCountry* with value set to the country's 3-letter ISO code (ex: USA).
4. Add a variable to that page type named *hr_CheckWidth* with value set to the size in inches of the check (default is 8.5).
5. Add a variable to that page type named *hr_CheckDPI* with value set to the resolution expected for the check processing engine (default is 300); typical is 200 to 300. You might need to adjust this value to get the best results, especially to recognize the MICR line.
6. Add field types with the symbolic names that are supported for the selected country. For example, for US checks:
 - o CheckNumber. Add a friendly name Check number
 - o Date
 - o Amount
 - o PayeeLine. Add a friendly name Payee
 - o MICRRoutineNumber. Add a friendly name Routine number
 - o MICRAccountNumber. Add a friendly name Account number

Adding the *ProcessCheck* variable also tells Datacap Mobile to expose the Check processing icon in the Properties screen so that the user can rerun recognition of the check, if needed. Check processing is invoked automatically at snap time if it is set as default.

Note that similar to ID recognition, there is no need to define zones for the server to grab the data. Analysis of the check image and extraction of the data are based on country-specific templates that are built in the Check Processing engine. This is why we need to pass the check country, size and DPI, so that the engine can select the template appropriate for the country and find the zones in the expected locations.

However you do need to assign the symbolic field names as specified by the engine. Datacap Mobile only displays the fields that are set up in the Document Hierarchy. For a US check, available fields are: CheckDate, CheckNumber, CheckSignature, Amount, Date, CAR, LAR, PayeeLine, MICR, PayorBlock, and PayorLine. Also, you can access the breakdown of the check Date into Month, Date, and Year, and the breakdown of the MICR line into its components MICRRoutineNumber, MICRAccountNumber, MICRCheckNumber, or MICRData0, MICRData1, and MICRData2, typically for getting at the routing, account, and check numbers.

Configuring the Payee Line field

Cursive and hand-print writing can be recognized in the Payee Line. However, you need to supply a dictionary of possible combinations of predefined vocabulary and aliases, each assigned a parameter that reflects the probability of occurrence in the image snippet of the Payee line. For example, you can define a list of the possible combinations around your company name (the name of the company the checks should be made payable to). Other payees are unlikely if you are not processing checks for the benefit of others. The recognition engine selects the vocabulary entries that are most probable. The path to the dictionary needs to be defined. By Default, in the Transaction application, which is the one invoking check recognition for Datacap Mobile, it is located under:\Datacap\Transaction\dco_Transaction\Vocabularies\payee.txt.

Here is an example:

```
"THE PAYMENT CENTER", 5
"PETER SMITH", 5
"HARBOR STREET FINANCIAL", 5
"HARBORSTREET FINANCIAL", 5
```

Where the number represents the probability weight from 1 to 10.

Configuring the geolocation

Datacap Mobile transfers the current location of the device to the Datacap application and images.

Procedure

To configure the geolocation, complete the following steps.

1. Make sure Location Services are turned on and authorized for the app on the device.
2. Use Datacap Studio or FastDoc to add a field with the symbolic name Location to any document or page that you want Datacap Mobile to store geolocation information to. It also stores the GPS latitude and longitude coordinates at the page level in *Latitude* and *Longitude* variables respectively, and in the JPEG image as EXIF geolocation tags to be used outside of the ECM repository.

Parent topic: [Configuring a Datacap application for Datacap Mobile](#)

Exporting encryption keys

You must generate security keys that allow Datacap to encrypt and decrypt passwords. To replace existing keys with new keys, you can specify a parameter to generate encryption keys to a local store. You can specify a different parameter to export the encryption keys from the local keystore to a file that can be imported to other computers.

About this task

You must generate and use the security encryption keys that allow Datacap to encrypt and decrypt the passwords that are used by users, services, and processes to access the Datacap server service and to log in to databases.

In a single machine configuration, you must generate and export the encryption keys that all of the Datacap components on the single machine use.

In a client/server configuration, you must generate and export matching security encryption keys from the server on which the Datacap server software component is installed to all of the computers on which any Datacap component is installed. This requirement secures any passwords that are passed over or received from the network by the Datacap component.

Procedure

To generate encryption keys and export them:

1. Open a command prompt and navigate to the C:\Datacap\Taskmaster folder. In a client/server configuration, perform this step on the computer on which the Datacap server software component is installed.
2. Run the key management program, `dcskey.exe`, inserting one or more of the following parameters in the command. For example, to export keys during a new Datacap installation, you would enter `dcskey.exe`

`e`

Exports the encryption keys from the local keystore to a `dc_KTF.xml` key transport file. You can use this file to import the keys to other computers. If no keys exist in the keystore, the `e` parameter generates new ones before the export. If keys exist in the keystore, the `e` parameter exports those keys.

`gnk`

Generates, but does not export, encryption keys in the local keystore. Use this parameter any time you must replace existing keys with new keys. For example, you would run the command

`dcscopy.exe gnkey` to replace existing keys and export them. The newly exported keys would then must be imported onto all other Datacap computers in your configuration.

Parent topic: [Installation instructions for Datacap server](#)

Client/server environment: Datacap Web Client installation and configuration

You can run tasks from a computer on which the Internet Explorer browser is installed when you set up the Datacap Web Client. In a single machine configuration that is used for testing and demonstration purposes, Datacap Web Client runs using the LocalSystem account.

The steps that you must follow to install and configure Datacap Web Client include ensuring that the required IIS components are installed and creating the Datacap Web Client site. You use the Datacap Web Client Server Configuration tool to set up the Datacap Web Client site for a single machine configuration. A single machine configuration requires the `tmweb.net` application pool to be set for the LocalSystem account.

- [Client/server environment: Installing and configuring Datacap Web Client on an IIS server](#)
You can install and configure Datacap Web Client on a Windows IIS server to use the browser-based Datacap Web Client for processing.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Client/server environment: Installing and configuring Datacap Web Client on an IIS server

You can install and configure Datacap Web Client on a Windows IIS server to use the browser-based Datacap Web Client for processing.

- [Client/server environment: Verifying that IIS components are installed](#)
Ensure that the IIS Web Server Role Services are installed for the Datacap Web Client server components, such as Datacap Web Client, Report Viewer, Fingerprint Service, and Datacap Web Services.
- [Client/server environment: Ensuring an account exists for Datacap Web Client](#)
Ensure that a domain/Windows account exists for Datacap Web Client. Datacap does not require a unique Windows account to be set up for Datacap Web Client.
- [Client/server environment: Installing Datacap Web Client](#)
You can install Datacap Web Client on the server using the Datacap installation program wizard.
- [Client/server environment: Importing encryption keys to Datacap computers](#)
In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.
- [Client/server environment: Creating the Datacap Web Client site](#)
You can create the Datacap Web Client site to run tasks from a computer on which the Internet Explorer browser is installed. The steps that you follow to install and configure the Datacap Web Client site include ensuring that the required IIS components are installed and configuring the `tmweb.net` application pool.
- [Client/server environment: Setting the Datacap Web Client Application Pool Identity](#)
You must set the Datacap Web Client Application Pool Identity in the Internet Information Services (IIS) Manager window.
- [Client/server environment: Changing the SSL setting in the server.ini file \(optional\)](#)
When you use Secure Socket Layer to encrypt communications between Datacap Web Client and its

clients, you can change the value of the SSL setting in the server.ini file.

Parent topic: [Client/server environment: Datacap Web Client installation and configuration](#)

Client/server environment: Verifying that IIS components are installed

Ensure that the IIS Web Server Role Services are installed for the Datacap Web Client server components, such as Datacap Web Client, Report Viewer, Fingerprint Service, and Datacap Web Services.

About this task

The Datacap Web Client server components must have specific IIS Web Server Role Services installed.

Procedure

Ensure that the IIS components are installed.

1. From the Start menu on the web server, select Administrative Tools > Server Manager.
2. In the Server Manager hierarchy pane, expand Roles and select Web Server (IIS).
3. In the Web Server (IIS) pane, expand Role Services. Under Common HTTP Features ensure that Static Content, Default Document, Directory Browsing, and HTTP Errors are installed. If you are installing wTM, do not install the WebDAV Publishing role service because it prevents the Datacap web services PUT method from functioning.
4. In the Role Services pane under Application Development, ensure that the ASP.NET, .NET Extensibility, ASP, ISAPI Extensions, and ISAPI Filters are installed.
5. In the Role Services pane, under Health and Diagnostics ensure that HTTP Logging and Request Monitor are installed.
6. In the Role Services pane, under Security ensure that Request Filtering is installed.
7. In the Role Services pane, under Performance ensure that Static Content Compression is installed.
8. In the Role Services pane, under Management Tools ensure that IIS Management Console is installed.
9. (Windows Server 2012 R2) In Server Manager, click Dashboard, and click Add roles and features.
 - a. In the Add Roles and Features Wizard, go to the Features section.
 - b. In .NET Framework 4.5 Features > WCF Services, select the HTTP Activation check box.
10. Close the Server Manager window.

Parent topic: [Client/server environment: Installing and configuring Datacap Web Client on an IIS server](#)

Client/server environment: Ensuring an account exists for Datacap Web Client

Ensure that a domain/Windows account exists for Datacap Web Client. Datacap does not require a unique Windows account to be set up for Datacap Web Client.

About this task

Datacap Web Client can use any Windows account that is set up with the appropriate sharing and security permissions. When Datacap Web Client and Report Viewer are installed on the same web server, they must use the same domain/Windows account. Be sure that you set up the appropriate sharing and security permissions for the domain/Windows account used by Datacap Web Client. These permissions were included as part of the installation instructions for the Datacap server.

Client/server environment: Installing Datacap Web Client

You can install Datacap Web Client on the server using the Datacap installation program wizard.

Before you begin

Before installing Datacap Web Client, [stop or ensure the Datacap Server Service](#) is stopped.

About this task

This procedure provides instructions on how to run the Datacap installation program wizard on a server to install the Datacap software components. The components that you install are the Datacap Web Server component and the separately licensed Datacap connectors for which you have a license.

Procedure

1. Make the installation package available on your network or insert the Datacap CD in the Server's CD/DVD drive. If the installation process does not start automatically, or if the package is on the network, open Windows Explorer, navigate to and double-click the Setup.exe. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.
2. Select the appropriate language and click OK. The language that you select determines the language that is displayed by the installation program wizard during the installation process.
3. When additional, redistributable software is required, the installation program wizard displays a list of the items to be installed. Click Install.
4. Click Next.
5. Accept the license agreement and click Next.
6. Select the Custom option and click Next.
7. Exclude all components from the installation process except Datacap Web Server and the separately licensed connectors for which you have a license.
8. Click Next.
9. Click Install.
10. Click Finish.

Parent topic: [Client/server environment: Installing and configuring Datacap Web Client on an IIS server](#)

Client/server environment: Importing encryption keys to Datacap computers

In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.

Before you begin

You must generate the encryption keys in the keystore on a server on which the Datacap server software component is installed. You export the new keys to a key transport file.

Procedure

To import encryption keys to Datacap computers:

1. Find the dc_KTF.xml key transport file in the c:\Datacap\Taskmaster folder on the Datacap server where you generated and exported the encryption keys.
2. Copy the dc_KTF.xml key transport file to the appropriate folder on the computer where you installed the new component. The encryption keys are automatically applied to the keystore the next time you start the Datacap component.

Table 1. Encryption key folder locations by component

Component	Folder
Datacap Server	C:\Datacap\Taskmaster
Datacap applications	C:\Datacap\Taskmaster
Datacap Desktop	C:\Datacap\DcDesktop
Datacap Studio	C:\Datacap\DStudio
Datacap FastDoc	C:\Datacap\FastDoc
Datacap Report Viewer	C:\Datacap\RV2\bin
Datacap Web Client	C:\Datacap\tmweb.net\bin
Datacap Web Services	C:\Datacap\wTM\bin

Parent topic: [Client/server environment: Installing and configuring Datacap Web Client on an IIS server](#)

Client/server environment: Creating the Datacap Web Client site

You can create the Datacap Web Client site to run tasks from a computer on which the Internet Explorer browser is installed. The steps that you follow to install and configure the Datacap Web Client site include ensuring that the required IIS components are installed and configuring the tmweb.net application pool.

Before you begin

Before creating the Datacap Web Client site, [stop or ensure the Datacap Server Service](#) is stopped.

About this task

You must create the Datacap Web Client site by using the Datacap Web Client Server Configuration tool.

Procedure

1. From the Windows Start menu, select IBM Datacap Web > Datacap Web > Server Configuration Tools. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.
2. Determine whether the contents of the information box indicate one of the following conditions:
 - o If one or more of the items that are listed in the information box are Not Found, click OK, then follow the instructions in [Microsoft Internet Information Services and Microsoft .NET Framework](#) to determine whether IIS and .NET were installed in the correct sequence and that all of the IIS components are installed.
 - o If all of the items that are listed in the information box are Found, click OK and continue with these instructions.

3. Ensure that the default value for each of the settings is appropriate for your web site, processing needs, and corporate requirements.
 - o When your Datacap Web Client site is not nested under the Default Web Site, select the appropriate Site and change the Site Settings.
 - o Depending on the usual size of the images or groups of images that are uploaded from the Datacap Web Client, you must adjust the value of the ASP.NET Maximum File Upload Size and ASP Maximum Requesting Entity Body Limit fields.
 - o Depending on your typical processing loads and schedules, change the default value of the App Pool Recycling Schedule to a time during which there are few or no batches in process Datacap Web Client. For more information on recycling and Datacap Web Client, see *Effects of IIS application pool recycling on Datacap Web Client batches*.
 - o Depending on your company's security policies, change the value of the Connection time-out field.
4. Click Configure.
5. Click OK, then click Exit. The Datacap Web Client site is created.
6. From the Windows Start menu, select Control Panel > Administrative Tools, then double-click Internet Information Services (IIS) Manager.
7. In the Connections pane, expand the computer node, expand the Sites node, and expand the Default Web Site or your web site. The tmweb.net site is displayed. If it is not displayed, right-click the site and select Refresh.
8. In the Application Pools pane, select the tmweb.net application pool, then in the Actions pane, in the Edit Application Pool section, click Advanced Settings.
9. In the Process Model section, set Load User Profile to True.
10. Click OK.
11. In the Connections pane, expand the Sites node, and expand the Default Web Site or your web site.
12. Select the tmweb.net site, and in the middle pane, double-click Session State.
13. Under Cookie Settings, change the Name to `tmweb` or another unique name, then, in the Actions pane, click Apply.
14. In the Connections pane, select the Default Web Site or your web site, then, in the Actions pane, under Manage Web Site, click Restart.

Parent topic: [Client/server environment: Installing and configuring Datacap Web Client on an IIS server](#)

Client/server environment: Setting the Datacap Web Client Application Pool Identity

You must set the Datacap Web Client Application Pool Identity in the Internet Information Services (IIS) Manager window.

About this task

This procedure provides instructions on how to set the Application Pool Identity in IIS that Datacap Web Client uses to access Datacap.

Procedure

To set the Datacap Web Client Application Pool Identity:

1. From the Windows Start menu, select Control Panel > Administrative Tools > Internet Information Services Manager.
2. In the Connections pane, expand the server node and select Application Pools.

3. In the Application Pools pane, select tmweb.net AppPool, then in the Actions pane, in the Edit Application Pool section, click Advanced Settings.
4. In the Process Model section, click the Browse button next to the Identity field.
5. In the Application Pool Identity window, select Custom account and click Set.
6. In the Set Credentials window, enter the Datacap Web Client domain/Windows account information in the format, `accountname@domainname`. Enter the password twice, then click OK.
7. In the Process Model section, set Load User Profile field to True.
8. Click OK.
9. Confirm that WebServer, Application Pool, and Default Web Site are started.

Parent topic: [Client/server environment: Installing and configuring Datacap Web Client on an IIS server](#)

Client/server environment: Changing the SSL setting in the server.ini file (optional)

When you use Secure Socket Layer to encrypt communications between Datacap Web Client and its clients, you can change the value of the SSL setting in the server.ini file.

Procedure

1. On the Datacap Web Client server, start Windows Explorer, navigate to, and use a text editor, such as Notepad, to open the `C:\Datacap\tmweb.net\server.ini` file.
2. Change the `UseSSL=0` setting to `UseSSL=1`, save the change, and close the server.ini file.

Parent topic: [Client/server environment: Installing and configuring Datacap Web Client on an IIS server](#)

Installing and configuring Datacap Navigator

Datacap Navigator is a web client for Datacap based on IBM® Content Navigator. You set up the Datacap Navigator client by loading a plug-in and configuring repositories and desktops in IBM Content Navigator.

To set up Datacap Navigator, you first install the client as a plug-in to IBM Content Navigator. Then, you configure repositories that correspond to Datacap applications. Finally, you can customize the default desktops that are generated for Datacap Navigator.

- [Datacap Navigator installation steps](#)
Datacap Navigator is installed as a plug-in to IBM Content Navigator.
- [Upgrading IBM Daeja ViewONE Virtual](#)
IBM Daeja® ViewONE Virtual is included with IBM Content Navigator. If you plan to use IBM Daeja ViewONE Virtual to view documents in Datacap Navigator, you must use IBM Daeja ViewONE Virtual Version 4.1.5 with iFix001 or later.
- [Adding applications to Datacap Navigator](#)
Configure repositories to specify additional Datacap applications that will be available to users in the Datacap Navigator client.
- [Configuring variables in the batch structure](#)
By default, Datacap Navigator displays read-only TYPE and STATUS variables for batches in the batch structure (also known as the DCO Tree View). You can configure Datacap Navigator to display variables as drop-down lists that allow users to select values. Alternatively, you can display variables as text boxes that allow users to input values for each variable. You can also add and remove variables from the batch structure.
- [Creating a choice list for a field in the field detail panel](#)
In Datacap Navigator, you can add a page field in the field detail panel as a select box or a choice list.

- [Adding Datacap Navigator tasks to your application](#)
To work with an application in Datacap Navigator, the application must include Datacap Navigator tasks.
- [Customizing Datacap Navigator desktops](#)
You can customize Datacap desktops by modifying the default desktop settings.
- [Configuring Internet Explorer for TWAIN scanning in Datacap Navigator](#)
You can enable Web TWAIN scanning and document import in Google Chrome and Mozilla Firefox browsers by installing Web TWAIN. Datacap Navigator users are prompted to download and install a service component the first time they attempt to scan. By default, Microsoft Internet Explorer uses ActiveX for TWAIN scanning and document import. ActiveX is configured manually or by running the Web Client Configuration Tool. Web TWAIN scanning is more secure than using ActiveX.
- [Running validation rules](#)
You can run validation rules on Start Batch Panel fields.
- [Extracting data to drop-down list](#)
You can verify panes that can populate drop-down lists from line item details and in table cells.
- [Role-based redaction](#)
You can use the role-based redaction feature to hide specific sensitive information in a document or image.
- [Role-based batch filtering](#)
The Role-based batch filtering feature provides a security mechanism by restricting the access to the batches that depend on the authorization level of the user for a particular application, for example, TravelDocs. In Datacap Navigator, the users can only view and work on the batches based on the user's groups. This feature also enables Datacap Navigator users to assign group access control to batches by using standard actions. Role-based filtering is designed to work with any authentication method.
- [Creating custom panels in Datacap Navigator](#)
A panel is a data entry screen. Panels are dynamically generated by the system and require no additional setup. You can create your own layouts by creating custom panels for verify tasks, start batch panels, and batch editor panels. You can arrange the fields and change the appearance and behavior of the panels as required.
- [Adding a Start Panel widget on a Classify task page by using the Classify.js program](#)
You can use the Datacap administration settings to add the Start Panel widget on a Classify task page by using the Classify.js program. By default, the Classify task page displays the Start Panel widget between the Image Viewer and Batch Structure widgets. You can change the location of the widgets on the page.
- [Adding a Start Panel widget on a Multiple task page by using the Multiple program](#)
You can use the Datacap administration settings to add the Start Panel widget on a Multiple task page by using the Multiple program. By default, the Multiple task page displays the Start Panel widget between the Image Viewer and Batch Structure widgets. You can change the location of the widgets on the page.
- [Changing the layout of the widgets by changing your user settings](#)
You can change the location of the widgets to be displayed on the Classify or Multiple task page.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Datacap Navigator installation steps

Datacap Navigator is installed as a plug-in to IBM® Content Navigator.

Before you begin

Ensure that the following prerequisites for using Datacap Navigator are met:

- Datacap Web Services (wTM) are installed and configured. For instructions, see [Datacap Web Services installation steps](#).
- IBM Content Navigator Version 2.0.3 or later is installed and running.

- If you plan to use IBM Daeja® ViewONE Virtual to view documents in Datacap Navigator, ensure that IBM Daeja ViewONE Virtual version 4.1.5 with iFix001 is installed. For more information, see [Upgrading IBM Daeja ViewONE Virtual](#).
- Datacap applications are configured.
- You have administrator privileges on your scan workstation. The first time that you scan, you are prompted to download and run an installation program for Web TWAIN Scan. Workstation Administrator privileges are required to run the installation program. Alternatively, you can install Web TWAIN manually by using an MSI file: *Datacap_installation\tmweb.java\DynamicWebTWAINHTML5Edition.msi*. You can also deploy Web TWAIN to client computers by using a Windows Group Policy; for instructions, see <https://support.microsoft.com/en-us/kb/816102>.

Procedure

To install Datacap Navigator:

1. Access the IBM Content Navigator administration tool. In a browser, enter a URL with the following format:

```
http://host_name:port_number/context_root/?desktop=admin
```

By default, the context root is `navigator`.

2. Select Plug-ins and click the New Plug-in button.
3. In the JAR file path field, enter the location of the Datacap Navigator plug-in file: `DatacapWebPlugin.jar`. You can enter the URL or the fully qualified path of the plug-in file. By default, the plug-in file is installed in the `Datacap_install/tmweb.java/` directory. If the plug-in file can be accessed directly, enter the fully qualified path of the plug-in file. For example:

On Windows: `C:\datacap\tmweb.java\DatacapWebPlugin.jar`

On AIX: `opt/DatacapWebPlugin.jar`

If the plug-in file is published on a web server, you can enter the URL of the plug-in file.

Attention: Ensure that the plug-in file is not in your JDBC directory or in a directory that is included in the class path of your web application server. Otherwise, the plug-in will fail to load and the following error will be generated:

```
java.lang.NoClassDefFoundError: com.ibm.ecm.extension.Plugin
```

4. Click Load. The Datacap Client plug-in configuration page loads.
5. Optional: In the Application field, enter the name of the application to use with default Datacap desktops. Later, you can add more applications. For instructions, see [Adding applications to Datacap Navigator](#).
6. Optional: In the Datacap wTM URI field, enter the Datacap Web Services URI that will be used to connect to your Datacap server. For example: `http://9.126.73.125:808/ServicewTM.svc`.
7. Optional: Click Generate Default Desktop to create default Datacap desktops. To modify the default desktops or create more desktops, see [Customizing Datacap Navigator desktops](#).
8. Click Save and Close.
9. If Datacap Web Services are installed on IBM WebSphere® Application Server, restart WebSphere Application Server.
10. Validate that you can access Datacap Navigator. For more information, see [Datacap Navigator access](#).

Parent topic: [Installing and configuring Datacap Navigator](#)

Related information:

[IBM Content Navigator documentation](#)

Upgrading IBM Daeja ViewONE Virtual

IBM® Daeja® ViewONE Virtual is included with IBM Content Navigator. If you plan to use IBM Daeja ViewONE Virtual to view documents in Datacap Navigator, you must use IBM Daeja ViewONE Virtual Version 4.1.5 with

iFix001 or later.

About this task

If you are using IBM Content Navigator Version 2.0.3 with Fix Pack 6, install iFix001 or later for IBM Daeja ViewONE Virtual Version 4.1.5. To acquire a Daeja iFix:

- Download the Daeja Viewer iFix from Fix Central if you have the Daeja stand-alone license.
- Contact IBM Support to request the Daeja Viewer iFix if you have the Daeja embedded license.

For more information about Daeja licenses, see [Daeja fix integration with ECM client products](#).

If you are using IBM Content Navigator Version 2.0.3 with Fix Pack 4 or Fix Pack 5, you must first manually upgrade IBM Daeja ViewONE Virtual to Version 4.1.5 and then install iFix002 or later. To manually upgrade IBM Daeja ViewONE Virtual, see [Upgrading the Daeja ViewONE version on IBM Content Navigator V2.0.3](#). If you do not have a standalone license to IBM Daeja ViewONE Virtual, you must upgrade IBM Content Navigator to Fix Pack 6, and then contact IBM Support.

Procedure

To download and install iFix001:

1. Log in to [Fix Central](#).
2. In the Product Selector list, enter `Enterprise Content Management`.
3. In the Product list, select `Daeja ViewONE Virtual`.
4. In the Installed Version list, select `4.1.5`.
5. Select your platform and click `Continue`.
6. Browse to iFix001, download, and install. See the iFix001 Readme for installation instructions.

Parent topic: [Installing and configuring Datacap Navigator](#)

Adding applications to Datacap Navigator

Configure repositories to specify additional Datacap applications that will be available to users in the Datacap Navigator client.

Procedure

To configure repositories for Datacap Navigator:

1. Access the IBM® Content Navigator administration tool. In a browser, enter a URL with the following format:

```
http://host_name:port_number/context_root/?desktop=admin
```

By default, the context root is `navigator`.

2. Click `Repositories` in the left pane.
3. On the `Repositories` page, click `New Repository` and select `Datacap Application`. You might need to refresh your browser to view the `Datacap Application` option.
4. Specify the following fields:

Field name	Instructions
Display name	Enter the name of the Datacap application that will be displayed in Datacap Navigator.

Field name	Instructions
ID	The ID is generated automatically when you enter the display name.
Datacap wTM URI	Enter the web service entry point to the Datacap Web Services API.
Application	Select a Datacap application. It might take a few seconds for the list to be populated with application names. If the list is not populated, verify that the URI is correct and Datacap Web Services (wTM) are configured correctly. Then try again.
Use ActiveX in IE	Click Yes to use ActiveX for TWAIN scanning and document import in Internet Explorer. ActiveX is configured manually or by running the Web Client Configuration Tool. Click No to use Web TWAIN scanning in Internet Explorer. Web TWAIN scanning is more secure than ActiveX. Internet Explorer version 9 supports ActiveX only.
Use Virtual Viewer	Click Yes to use Daeja ViewONE Virtual to view documents in the repository. Daeja ViewONE Virtual is a server-based viewer that does not require Java on the client. Click No to use Daeja ViewONE Professional to view documents in the repository. Java must be installed on the client to use Daeja ViewONE Professional.

5. Click Connect and log in to the application.
6. Click the Configuration Parameters tab.
7. Specify whether to enable single sign-on. When you enable this option, you must specify LLDAP authentication in Datacap Server Manager. For more information, see [Configuring single sign-on \(SSO\) for Datacap Navigator](#) and [LLLDAP group authentication](#).
8. Select the columns that you want to display by default in Datacap Navigator on the Job Monitor and Task List pages.
9. Click Save and Close.
10. Optional: Repeat this procedure to make additional applications available in Datacap Navigator.

Parent topic: [Installing and configuring Datacap Navigator](#)

Configuring variables in the batch structure

By default, Datacap Navigator displays read-only TYPE and STATUS variables for batches in the batch structure (also known as the DCO Tree View). You can configure Datacap Navigator to display variables as drop-down lists that allow users to select values. Alternatively, you can display variables as text boxes that allow users to input values for each variable. You can also add and remove variables from the batch structure.

Procedure

To configure the display of variables in the batch structure:

1. Open the Datacap Navigator Administration View. In a browser, enter a URL with the following format:

```
http://host_name:port_number/context_root/?desktop=dcadmin
```

By default, the context root is `navigator`.

2. Navigate to the Display variables area of the advanced task settings.
 - a. In the left pane, click Workflows.
 - b. In the right pane, select a workflow and click Edit.
 - c. Select a job and click Edit.

- d. Select a task and click Edit.
 - e. On the Task > Advanced tab, scroll down to the Display variables area.
3. Enter key-value pairs in the Props for var fields to configure how variables are displayed in the batch structure. The first field contains the variable name (such as TYPE). Enter values in the second field as follows:

Value in Props for Var field (second box)	Description
0	Displays read-only variable values.
1	Displays editable variable values. A drop-down list is displayed for the TYPE and STATUS columns in the batch structure. When you enter 1 for other variables, a text box editor is displayed in the batch structure column for the variable.
1, setup	Displays a drop-down list of values that are configured in the Setup DCO dictionary. Use Datacap Studio to define the dictionary. The name of the dictionary is based on the level: Batch level The dictionary name is B:colname Document level The dictionary name is D:colname Page level The dictionary name is P:colname
1, comma-separated list of values	Specify values to display in the drop-down list. For example, enter 1, a, b, c to display a drop-down list that contains the values a, b, and c.

Tip: To add a variable, click the plus sign (+). To remove a variable, click the minus sign (-).

Parent topic: [Installing and configuring Datacap Navigator](#)

Creating a choice list for a field in the field detail panel

In Datacap Navigator, you can add a page field in the field detail panel as a select box or a choice list.

Procedure

To add the page field in the field detail panel as a choice list, complete the following steps:

1. Open the application in Datacap Studio
2. Select a page field and add *DICT* or *SELECT* variable in this field.
3. Login to Datacap Navigator again.

Note: Do not add "Text" or "Sticky" variable to this field. If this field has the variable "RecogType" with a value "4", then the field is displayed as a radio button or check box.

Parent topic: [Installing and configuring Datacap Navigator](#)

Adding Datacap Navigator tasks to your application

To work with an application in Datacap Navigator, the application must include Datacap Navigator tasks.

Before you begin

Ensure that the Datacap Navigator plug-in is deployed and a repository is created for the application.

About this task

When you create an application by using the Application wizard in Datacap Version 9.0 Feature Pack 2 or later, the application has Datacap Navigator tasks by default. If you want to work with an existing Datacap Version 9.0 application which does not have Datacap Navigator tasks, you must manually add these tasks. If you want to work with an application from a previous version, you must migrate the applications first; see the following topics for instructions:

- [Migrating Datacap applications from 8.0.1 to 9.0](#)
- [Migrating Datacap applications from 8.1 to 9.0](#)

Procedure

To manually add Datacap Navigator tasks to your application:

1. Open the Datacap Navigator Administration View. In a browser, enter a URL with the following format:

```
http://host_name:port_number/context_root/?desktop=dcadmin
```

By default, the context root is `navigator`.

2. In the left pane, click Workflows.
3. In the right pane, select a workflow and click Edit. Then, create a job and tasks for Datacap Navigator.
 - a. On the Task > General tab, specify Program options for Datacap Navigator tasks as follows:

Datacap Navigator task	Program option on the General tab
Scan	Scan.js
Upload	Upload.js
Verify	Multiple
Fixup	Multiple

- b. On the Task > Advanced tab, scroll down to the Datacap Navigator area and specify Web Program options for Verify and Fixup tasks as follows:

Datacap Navigator task	Web Program option on the Advanced tab
Verify	Verify.js
Fixup	Classify.js

Tip: You can configure Datacap Navigator tasks based on the task settings in the Datacap Accounts Payable (APT) sample application that is included with Version 9.0 Feature Pack 1 and later. View the task settings that are defined for the default Web Demo job in the APT workflow.

4. In the left pane, click Shortcuts. Create shortcuts for Datacap Navigator tasks.

Parent topic: [Installing and configuring Datacap Navigator](#)

Customizing Datacap Navigator desktops

You can customize Datacap desktops by modifying the default desktop settings.

Procedure

To customize Datacap Navigator desktops:

1. Access the IBM® Content Navigator administration tool. In a browser, enter a URL with the following format:

```
http://host_name:port_number/context_root/?desktop=admin
```

By default, the context root is `navigator`.

2. Click Desktops in the left pane.
3. Select a Datacap desktop to customize and click Edit.
4. On the General page in the Authentication area you can specify the repository (Datacap application) that is opened by default when you start Datacap Navigator. You can also limit access to specific users or groups.
5. Click the Repositories tab. You can specify the Datacap applications to expose in the desktop.
6. Click the Layout tab.
 - o In the Desktop Features area, you can add or remove access to IBM Content Navigator features. For example, select Browse to enable browsing in the repository from the Datacap Navigator interface. When you select the Browse feature, a Browse icon is added to the Datacap Navigator window in the left pane.
 - o In the Additional Desktop Components area, you can choose whether to show or hide document thumbnails, the global toolbar, and the status bar.
7. Click the Menus tab. In the Context Menus > Feature Context Menus area, ensure that Datacap User Session Context Menu is specified for the Banner user session context menu option. Specifying this option displays the Change User Settings option on the user ID drop-down menu in the Datacap Navigator banner.

Parent topic: [Installing and configuring Datacap Navigator](#)

Configuring Internet Explorer for TWAIN scanning in Datacap Navigator

You can enable Web TWAIN scanning and document import in Google Chrome and Mozilla Firefox browsers by installing Web TWAIN. Datacap Navigator users are prompted to download and install a service component the first time they attempt to scan. By default, Microsoft Internet Explorer uses ActiveX for TWAIN scanning and document import. ActiveX is configured manually or by running the Web Client Configuration Tool. Web TWAIN scanning is more secure than using ActiveX.

About this task

Internet Explorer version 9 supports ActiveX only. You can configure Internet Explorer versions 10 and 11 to use Web TWAIN scanning instead of ActiveX as follows:

Procedure

1. Access the IBM® Content Navigator administration tool. In a browser, enter a URL with the following format:

```
http://host_name:port_number/context_root/?desktop=admin
```

By default, the context root is `navigator`.

2. Click Repositories in the left pane.
3. On the Repositories page, select a Datacap application and click Edit.
4. On the General tab, select No for the Use ActiveX in IE setting.

Parent topic: [Installing and configuring Datacap Navigator](#)

Related information:

[IBM Content Navigator configuration database](#)

Running validation rules

You can run validation rules on Start Batch Panel fields.

Parent topic: [Installing and configuring Datacap Navigator](#)

Datacap Navigator Application configuration

Configure Datacap Navigator to run validation rules.

Procedure

To run validation rules on Start Batch Panel, the name of the validation task profile must be entered in the Datacap Navigator Application task settings:

1. Start Datacap Navigator, select the Administrator view, click Workflow, and select the Remote Scan Task for which you want to specify the task profile.
2. In the Selected task details pane, click the Advanced tab.
3. Scroll to the Rulerunner settings section.
4. In the Main task profile field, enter a name for the validation task profile, such as `ValidateStartPanel` as it was defined in Datacap Studio Application Task Profile.
5. Click Save.

Important: When you scan an image, if the "required" field in the Start Batch Panel is blank or empty, and the batch is submitted, a window opens that prompts the user to enter a value into the "required" field.

Datacap Studio configuration

Configure Datacap Studio to run validation rules.

Procedure

1. Start Datacap Studio.
2. Create a task profile. For example, "ValidateStartPanel" that is associated to ruleset.
3. Create a validation rule for each Start Batch Panel field that you want to validate.
4. Bind each rule to the associated field in the DCO. For example, the following rule validates if the field is a "required field": Validate Start Batch Panel ruleset:
 - `Status_Preserve_OFF()`
 - `SetIsOverrideable("false")`
 - `IsFieldFilled("FieldName")`

Batch-level metadata can be entered from a Start Batch panel and the display is based on the specific task settings. The Start Batch panel is dynamically created with data entry fields automatically displayed for all fields that are defined at the batch level within the application setup DCO.

Extracting data to drop-down list

You can verify panes that can populate drop-down lists from line item details and in table cells.

Support populating a drop-down list with a list of extracted values

If you run rule set and generate the extracted data in page data file, for example, tm000001.xml, you can add a variable "CHOICELIST" to define extracted values on a field.

The following three ways are supported to associate a field with the data in another field (consider this field as the field saving extracted data):

The extracted data should be in JSON format:

1. `<V n="CHOICELIST">{"mode": "r", "choices": [{"displayName": "International Business Machines Corporation", "value": "IBM"}, {"displayName": "Microsoft Corporation", "value": "MS"}]}</V>`
2. `<V n="CHOICELIST">{"mode": "r", "choices": ["IBM", "MS"]}</V>`
3. `<V n="CHOICELIST">{"mode": "a", "choices": "EntityList, Entity"}</V>`

mode: "r" -> replace the original choice list with the extracted data.

mode: "a" -> append the extracted data to the original choice list.

if no "mode", default is "a"

If "choices" is a string, the string format should be: "<field name>, <column name>" that means the string is the field to extract data, the field should be a table.

For example:

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.install.doc_dcnvi013_Text
_Field">
  <V n="TYPE">Text_Field</V>
  <V n="Position">582,101,926,176</V>
  <V n="CHOICELIST">{"mode": "r", "choices": ["IBM_1", "Microsoft_2", "Oracle_3",
"EMC_4"]} </V>

<V n="STATUS">0</V>
...
```

Support drop-down list box within a Datacap Navigator table cell

This supports the following JSON format:

1. `<V n="<TYPE of column>_CHOICELIST"V n="CHOICELIST">{"mode": "r", "choices": [{"displayName": "International Business Machines Corporation", "value": "IBM"}, {"displayName": "Microsoft Corporation", "value": "MS"}]}</V>`
2. `<V n="<TYPE of column>_CHOICELIST"V n="CHOICELIST">{"mode": "r", "choices": ["IBM", "MS"]}</V>`
3. `<V n="<TYPE of column>_CHOICELIST">{"mode": "r", "choices": "Details,Price"}</V><V n="CHOICELIST">{"mode": "a", "choices": "EntityList,Entity"}</V>`

For example:

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.install.doc_dcnvi013_Taxe
s">
<V n="TYPE">Taxes</V>
<V n="Position">0,0,0,0</V>
<V n="STATUS">0</V>
<V n="PreVerify Val"/>
<V n="PreVerify Pos">0,0,0,0</V>
<V n="label">Taxes</V>
<V n="Tax_Value_CHOICELIST">{"mode": "r", "choices": "Details,Price"}</V>
<F
```

```

id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.install.doc_dcnvi013_TaxLineitem1">
<V n="TYPE">TaxLineitem</V>
<V n="STATUS">0</V>
<V n="Position">0,0,0,0</V>
<V n="PreVerify Val"/>
<V n="PreVerify Pos">0,0,0,0</V>
<V n="label">Tax Lineitem</V>
<F>
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.install.doc_dcnvi013_TaxType">
<V n="TYPE">Tax_Type</V>
<V n="STATUS">0</V>
<V n="Position">0,0,0,0</V>
<V n="PreVerify Val">Sales</V>
<V n="PreVerify Pos">0,0,0,0</V>
<V n="label">Tax_Type</V>
<C cn="10" cr="0,0,0,0">51</C>
</F>

<F>
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.install.doc_dcnvi013_TaxValue">
<V n="TYPE">Tax_Value</V>
<V n="STATUS">0</V>
<V n="Position">2249,2126,2379,2164</V>
<V n="Locale Checked">Yes</V>
<V n="PreVerify Val">107.63</V>
<V n="PreVerify Pos">2249,2126,2379,2164</V>
<V n="label">Tax Value</V>
<C cn="10" cr="2253,2130,2265,2160">53</C>
<C cn="10" cr="2274,2130,2293,2160">48</C>
<C cn="10" cr="2297,2130,2317,2160">48</C>
<C cn="10" cr="2322,2156,2326,2160">46</C>
<C cn="10" cr="2331,2130,2351,2160">48</C>
<C cn="10" cr="2355,2130,2375,2160">48</C>
</F>
</F>
<F>
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.install.doc_dcnvi013_TaxLineitem2">
...

```

Parent topic: [Installing and configuring Datacap Navigator](#)

Role-based redaction

You can use the role-based redaction feature to hide specific sensitive information in a document or image.

The redaction feature finds sensitive information in a document or image, and creates rectangular redaction annotations to cover the sensitive information. When the documents are subsequently viewed, the sensitive information is hidden from unauthorized users. You can edit the redactions in Datacap Navigator, and can export them to FileNet P8. The redactions are displayed as rectangles, which can be opaque or semi-transparent, and visible and editable in the Datacap Navigator client, but not in Desktop, FastDoc, or TMWeb. Redactions can be exported to FileNet P8, and subsequently viewed in IBM® Content Navigator.

- [Enabling Role-based redaction](#)
Role-based redaction feature introduces a rectangular annotation that hides sensitive information on a document image. When a user views a document image, the redacted information is either hidden, or

made visible, depending on the permission granted to the user. To use the Role-based redaction feature, you must configure and enable the feature.

- [Role-based redaction settings](#)
You can use the Role-based redaction settings, where you can assign policy editors, redaction editors, specify redaction reasons, and so on. Redaction Reasons and Policies must be pre-defined in IBM Content Navigator, and imported into the Datacap application by using the Datacap Navigator administrator feature.
- [Creating and editing role-based redaction](#)
You can create and edit role-based redactions in Datacap Navigator, and export the redactions to FileNet P8 Any user in Datacap Navigator has permission to create and edit redactions that are imported or defined for that Datacap application.

Parent topic: [Installing and configuring Datacap Navigator](#)

Enabling Role-based redaction

Role-based redaction feature introduces a rectangular annotation that hides sensitive information on a document image. When a user views a document image, the redacted information is either hidden, or made visible, depending on the permission granted to the user. To use the Role-based redaction feature, you must configure and enable the feature.

Before you begin

Ensure that the following prerequisites are installed to configure and use the Role-based redaction feature:

- IBM® Daeja® ViewONE Virtual 5.0, or later
For more information about configuring IBM Daeja ViewONE Virtual, see [Configuring the IBM Daeja ViewONE Virtual viewer](#).
Note: You must acquire a separate license for IBM Daeja ViewONE viewer redaction services. For more information, see [Adding entitlements to an integrated version of IBM Daeja ViewONE, V4.1 or later](#).

You can verify that whether virtual viewer license file is applied or not at this location: ...
`\ECMClient\configure\explodedformat\navigator\WEB-INF\lic-server-virtual.v1`

- IBM Content Navigator version 3.0, or later
Configure the IBM Content Navigator desktop to use a viewer mapping that assigns the IBM Daeja ViewONE Virtual viewer as the viewer for PDF and TIFF files for IBM FileNet P8 repositories.
- A FileNet P8 repository with Role-based redaction reasons and policies defined using IBM Content Navigator
- Datacap Navigator

Datacap Navigator is the only Datacap UI capable of displaying Role-based redactions, and creating or editing them.

Procedure

To enable and use role-based redaction, complete the following steps:

1. Log in to IBM Content Navigator.
2. In the navigation tree, click Viewer Map, and then click New Viewer Map to create a new viewer map with the name of your choice. For example, Virtual Viewer.

3. After adding the required details, click **New Mapping**, and specify the Repository type as "FileNet Content Manager" and Viewer as "Daeja ViewONE Virtual".
4. Specify the file type as "application/pdf", "image/tiff", and "image/x-tiff", and click **OK**.
Important: Single-page TIFF or JPG images can be exported with redactions. Exporting multi-page TIFF or PDF is not supported for redactions. You must split multi-page image files after ingestion. You might re-merge the files before exporting. When the `FNP8_Upload` action is used, redactions are saved in a FileNet P8 repository. Redactions are not currently preserved by the `FNP8_UploadDir` action.
5. Go to IBM Content Navigator repository configuration, and click **Configuration parameter**.
6. Click **Enable** for Role-based redaction.
Note: The supporting P8 add-on (IBM Content Navigator 3.0.0 Redaction Extensions) is installed when Role-based redactions option is enabled. The add-on cannot be removed after it is installed; however, you can disable the role-based redaction feature.

For more information about Role-based redaction settings, see [Role-based redaction settings](#)

Parent topic: [Role-based redaction](#)

Role-based redaction settings

You can use the Role-based redaction settings, where you can assign policy editors, redaction editors, specify redaction reasons, and so on. Redaction Reasons and Policies must be pre-defined in IBM® Content Navigator, and imported into the Datacap application by using the Datacap Navigator administrator feature.

Role-based redaction repository settings

Assign policy editors

- A policy editor is allowed to create, edit, and delete policies and roles.
- The creator of a policy or role is not granted any direct permissions to the policy or role. The creator has edit access only if they are a policy editor.
- Policies and roles are edited in the IBM Content Navigator administration interface so policy editors must also be IBM Content Navigator administrators.

Assign redaction editors

- A redaction editor is granted permission in the repository to create instances of the Role-based redaction annotation class.
- When the redaction editor user has permission to create annotations for the document, the final permission to create a Role-based redaction for a particular reason is controlled by redaction policies.
- The creator of a Role-based redaction is not granted any direct permissions to the redaction. Edit access is granted only by redaction policies.

Note: Redacted PDF files are delivered as PDF (default) or TIFF files.

Role-based Redaction Settings

Reasons

Redaction reason identifies the type of sensitive data that is redacted, and manages redaction reason definitions. A redaction reason:

- Has a Name, Description, and unique ID (GUID).
- Is assigned to each role-based redaction annotation.
- Is connected to role permissions by a policy.

- Redaction reason definitions are stored in the IBM Content Navigator configuration database and might be used in any FileNet P8 repository that is enabled for Role-based redaction.
- Datacap imports the redaction reason definitions from IBM Content Navigator.
- IBM Content Navigator maintains a record of whether a reason is used in Datacap or in a FileNet P8 repository. A reason that is in use cannot be deleted.
- Redaction reason definitions can be exported and imported by IBM Content Navigator.

Policies and Roles

- The Policies and Roles define user and group access for role-based redaction reasons, and are saved in the FileNet P8 repository.
- The Policies and Roles are managed by the Policy editors who are selected in the repository configuration.

Parent topic: [Role-based redaction](#)

Creating and editing role-based redaction

You can create and edit role-based redactions in Datacap Navigator, and export the redactions to FileNet P8. Any user in Datacap Navigator has permission to create and edit redactions that are imported or defined for that Datacap application.

Procedure

1. Log in to Datacap Navigator desktop of a Datacap application that implements role-based redactions.
2. Scan or import a pre-scanned image from a directory, and then click Submit.
The batch might be uploaded to the server automatically; otherwise, you can use the Upload shortcut to upload. The Job Monitor shows a table that contains the batch that you created.
Note: The Task column in the table shows the value as "Batch Profiler", and the Status column shows the value as "pending".
3. To refresh the status, click the Refresh icon. If Rulerunner Server is configured and active, the status changes to "running".

If Rulerunner is not running, you can run the Profiler task using Datacap Desktop thick client. The Profiler rules typically locate the sensitive information by using the DocumentAnalytics actions, and redact the information using the RedactFields action.

4. Click the Refresh icon. The Task column in the table shows the value as "Verify".
5. Ensure that the desired batch is selected, and then click Start.
The image is displayed overlaid with redactions over the sensitive data that are created by the automated process. The DocumentAnalytics actions detect the sensitive information in the image, and redact the information with rectangles. The rectangles represent redaction annotations.
Note: You can edit the redactions in Datacap Navigator, and select the information that you want to redact in the image.
6. Go to one of the redactions, right-click the redaction annotation, select the redaction reason, and then click Apply.

You can change the size or position of a redaction. You can also delete or create a new redaction by selecting the rectangle annotation tool in the left toolbar and dragging the mouse on the image.

7. When you are satisfied with the redaction positions and reasons, click Submit to submit the document to the next stage of the process, which is to export the document with redactions to FileNet P8.
8. Open the document in the IBM® Content Navigator viewer.

If you have permissions according to the role-based redaction policies in FileNet P8, then you can create privileged users to see the sensitive information and you can also make modifications, such as changing the size of the redaction. If you have no permissions, then the redacted information is always covered. Note: If a redaction is used in Datacap batches, the redaction persists after you export the redactions to FileNet P8.

If any redactions are created in a Datacap document (manually in verify, or automatically using RedactFields), and the doc is exported as one or more single-page TIF or JPG files, the redactions are uploaded too. In FileNet P8, redactions are considered a type of "annotation". In addition, if any pages in the document were processed (meaning, the OCR results were searched for sensitive content) using the RedactFields action, an audit record is created in FileNet P8, which is also considered a type of annotation. In addition, in FileNet P8 the document's Sensitive Content property is set to "1" if any redactions were created, and "0" if RedactFields was run but no sensitive data was found to redact. The P8 export / upload will fail if any of the following is true:

- the export repository does not have role based redactions enabled
- any of the redaction reasons (that were imported originally into the Datacap application from IBM Content Navigator) don't exist in the export FileNet P8 repository
- any of the redaction reasons don't have an associated redaction policy in the export FileNet P8 repository

Parent topic: [Role-based redaction](#)

Role-based batch filtering

The Role-based batch filtering feature provides a security mechanism by restricting the access to the batches that depend on the authorization level of the user for a particular application, for example, TravelDocs. In Datacap Navigator, the users can only view and work on the batches based on the user's groups. This feature also enables Datacap Navigator users to assign group access control to batches by using standard actions. Role-based filtering is designed to work with any authentication method.

There are the following modes that belong to the role based batch filtering feature:

- Exclusive - The batch must belong to one of the user's groups.
- Additive - The batch can belong to multiple groups but must belong to at least one of the user's groups.
- Inclusive - The batch can belong to multiple groups and user must belong to all those groups. User might belong to more groups.
- none - You can use this mode to disable the role based batch filtering feature.

Here are a few examples:

- A user can only view and work on the documents that are created on the user's current workstation. This is accomplished by assigning the station IDs when the user logs in to Datacap Navigator.
- When a user logs in, the user's access is limited to the batches that are currently owned by the LDAP group of that location.
- Users at Location A can scan a batch and make it quickly available for the first-pass verification at the same location.
- Users at Location A can see a list of batches in the Job Monitor that are tagged with the LDAP group for that Location A.
- Users at a particular location are authorized to view the batches that originate from that location. In addition, they can only run the tasks that are available in the Shortcut pane.
- [Weight for the groups](#)
You can assign weight to a particular group when you use the role-based batch filtering feature. A group with the highest weight, which is represented by 0 to 10, has priority over the other groups.

- [Assigning weight to the groups](#)
You can assign weight to a particular group when you use the role-based batch filtering feature.

Parent topic: [Installing and configuring Datacap Navigator](#)

Weight for the groups

You can assign weight to a particular group when you use the role-based batch filtering feature. A group with the highest weight, which is represented by 0 to 10, has priority over the other groups.

The following scenario explains the feature:

- There are the following two groups:
 - Group A: Weight assigned is 1
 - Group B: Weight assigned is 6
- Group A has the following two users:
 - User 1
 - User 2
- Group B has the following users:
 - User 2
 - User 3
- User 2 (Belongs to Group A) logs in to Datacap Navigator, and creates a batch.
- The batch that User 2 created belongs to the Group B as the Group B has higher weight than the Group A.

For more information about assigning weight to the groups, see [Assigning weight to the groups](#).

Parent topic: [Role-based batch filtering](#)

Assigning weight to the groups

You can assign weight to a particular group when you use the role-based batch filtering feature.

Procedure

To assign weight to a group, complete the following steps:

1. Log in to Datacap Navigator.
2. In the left pane, click Groups.
3. On the Groups page, select a group, and then click the Edit tab.
4. In the Weight list, select a number.
5. Click Save and Close.

Parent topic: [Role-based batch filtering](#)

Creating custom panels in Datacap Navigator

A panel is a data entry screen. Panels are dynamically generated by the system and require no additional setup. You can create your own layouts by creating custom panels for verify tasks, start batch panels, and batch editor panels. You can arrange the fields and change the appearance and behavior of the panels as required.

About this task

The following procedure describes how to create a custom panel for the verify task. The procedure for creating custom start batch and batch editor panel is similar.

Procedure

To create a custom verify panel:

1. Open the Datacap Navigator Administration View. In a browser, enter a URL with the following format:

```
http://host_name:port_number/context_root/?desktop=dcadmin
```

By default, the context root is `navigator`.

2. Click Panels in the left pane.
3. Click New Panel and select Verification Panel.
4. Specify the associated workflow, page type, and a name for the panel. A default panel layout is generated and displayed in the center pane. The default panel layout contains all fields that are defined for the selected page type.
5. Arrange fields in the panel. You can change the order by dragging fields in the center pane, remove any unwanted fields, and drag removed fields back into the center pane.
6. Click Save and Close.
7. Enable the new verify panel in the workflow.

Tip: For start batch panels, set the DCO Type in the Scan task advanced properties to `TravelDocs`. For batch editor panels, there is no need to configure task settings; the system fetches the last batch editor panel that was created for the application.

- a. Click Workflows in the left pane.
 - b. Select your workflow and click Edit.
 - c. Select Navigator Job and click Edit.
 - d. Click Tasks.
 - e. Select Verify and click Edit.
 - f. Click Advanced.
 - g. In the Custom web panels area, select Use custom web panels.
 - h. Enter the page type and panel name that you specified when you created the panel.
 - i. Click Save and Close to save your changes to the verify task.
8. Log out and close your browser. Your new panel is displayed the next time that you run the verify task.

- [Adding custom code to Datacap Navigator panels](#)

You can add functionality or UI elements to Datacap Navigator panels. For example, you can add a logo image to a specific panel by modifying the HTML code in the panel data file.

Parent topic: [Installing and configuring Datacap Navigator](#)

Adding custom code to Datacap Navigator panels

You can add functionality or UI elements to Datacap Navigator panels. For example, you can add a logo image to a specific panel by modifying the HTML code in the panel data file.

Procedure

To add custom code to a panel:

1. On the computer where Datacap is installed, use Windows Explorer to navigate to the panels folder. The panels folder is in the Datacap applications installation folder, for example:
C:\Datacap\APT\navigatorpanel\panels.
2. Open the panel (JSON file) that you want to modify in a text editor.

3. Modify the panel code. The JSON file is a standard Dojo widget template. For example, to add an element to the panel such as a logo image, add the URL of the logo image to the panel code: ``
4. Save and close the JSON file.

Parent topic: [Creating custom panels in Datacap Navigator](#)

Adding a Start Panel widget on a Classify task page by using the Classify.js program

You can use the Datacap administration settings to add the Start Panel widget on a Classify task page by using the Classify.js program. By default, the Classify task page displays the Start Panel widget between the Image Viewer and Batch Structure widgets. You can change the location of the widgets on the page.

Procedure

1. Log in to Datacap Navigator, and select an application. For example, TravelDocs.
2. Open the Datacap administration view.
3. Click the Workflows tab, and select a workflow.
4. On the Workflow page, click the Jobs tab, and select a Fixup job.
5. On the Job page, click the Tasks tab, and select a task. For example, FixUp.
6. On the Task page, under the General tab, the Program list displays a new value as `Classify.js`, select `Classify.js`.
7. Click the Advanced tab.
8. Select the Show the Start Batch Panel check box. When you select this check box, then after opening any Classify task, the task page shows the Start Panel.
9. In the Custom web panels area, select Use custom web panels check box, and enter the page type and panel name that you specified when you created the panel. For more information about creating custom panels, see [Creating custom panels in Datacap Navigator](#).
10. Click Save and Close.
11. Click the Layout tab. In the layout, by default, you can see the Start Panel widget between Image Viewer and Batch Structure widgets for the classify.js program. You can change the location by dragging and dropping the widget.
12. Open a Classify task, the Classify task page shows the Start Panel. For more information about changing the layout of the widgets by changing your user settings, see [Changing the layout of the widgets by changing your user settings](#)

Parent topic: [Installing and configuring Datacap Navigator](#)

Adding a Start Panel widget on a Multiple task page by using the Multiple program

You can use the Datacap administration settings to add the Start Panel widget on a Multiple task page by using the Multiple program. By default, the Multiple task page displays the Start Panel widget between the Image Viewer and Batch Structure widgets. You can change the location of the widgets on the page.

Procedure

1. Log in to Datacap Navigator, and select an application. For example, TravelDocs.
2. Open the Datacap administration view.
3. Click the Workflows tab, and select a workflow.

4. On the Workflow page, click the Jobs tab, and select a Fixup job.
5. On the Job page, click the Tasks tab, and select a task. For example, FixUp.
6. On the Task page, under the General tab, the Program list displays a new value as `Multiple`, select `Multiple`.
7. Click the Advanced tab.
8. Select the Show the Start Batch Panel check box. When you select this check box, then after opening any Multiple task, the task page shows the Start Panel.
9. In the Custom web panels area, select Use custom web panels check box, and enter the page type and panel name that you specified when you created the panel. For more information about creating custom panels, see [Creating custom panels in Datacap Navigator](#).
10. Click Save and Close.
11. Click the Layout tab. In the layout, by default, you can see the Start Panel widget between Image Viewer and Batch Structure widgets for the Multiple program. You can change the location by dragging and dropping the widget.
12. Open a Multiple task, the Multiple task page shows the Start Panel. For more information about changing the layout of the widgets by changing your user settings, see [Changing the layout of the widgets by changing your user settings](#)

Parent topic: [Installing and configuring Datacap Navigator](#)

Changing the layout of the widgets by changing your user settings

You can change the location of the widgets to be displayed on the Classify or Multiple task page.

Procedure

1. To change the user settings from any Datacap Navigator view, click the user ID menu, and then click Change User Settings.
2. On the Settings page, click the Classify tab, and then click the Layout tab.
3. Select the Customize the layout check box.
4. Change the layout by dragging and dropping the widget. The Classify page shows the widget location per your settings on the Layout page.

Parent topic: [Installing and configuring Datacap Navigator](#)

Installing the developer workstation software components

Run the Datacap installation program wizard on the workstation of a developer to install the Datacap software components.

About this task

The software components that you install include the Datacap client and sample applications. Also, the separately licensed applications, and connectors for which you have a license, the Datacap Studio, FastDoc, and Maintenance Manager software components.

These instructions apply to workstations that run on Windows 7.

Procedure

To install the developer workstation components:

1. Put the installation package on your network, or insert the Datacap CD in the developer workstation CD/DVD drive. If the installation process does not start automatically, or if the package is on the network, open Windows Explorer, go to and double-click the Setup.exe. Click Yes at the User Account Control window.
2. Select the appropriate language and click OK. The language that you select is displayed on the installation program screens during the installation.
3. When more, redistributive software is required, the installation program wizard displays a list of the items that must be installed, click Install.
4. Click Next.
5. Click to accept the license agreement and click Next.
6. Select the Custom option and click Next.
7. Exclude all of the components except the Datacap Client and the separately licensed applications and connectors for which you have a license. Be sure the FastDoc, Datacap Studio, and Maintenance Manager components are included.
8. Click Next.
9. Click Install.
10. Click Finish.

- [Creating or ensuring accounts exist for developers](#)

Datacap does not require that a unique Windows account is set up for each developer. A developer can use any Windows account that is configured with the appropriate sharing and security permissions.

- [Importing encryption keys to Datacap computers](#)

In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.

- [Setting up the scanner](#)

When a computer is to be used as a scan station, follow the manufacturer's instructions to attach the scanner to the computer. Install and configure the required scanner drivers and software. Be sure you can scan successfully with an image capture software product other than Datacap. If you are unable to scan documents with software that is not Datacap, you cannot scan documents with Datacap.

- [Configure Internet Explorer on the developer workstation](#)

You must configure Internet Explorer on the developer workstation to enable access to Datacap Web Client, the system administration panel, and the user panel.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Creating or ensuring accounts exist for developers

Datacap does not require that a unique Windows account is set up for each developer. A developer can use any Windows account that is configured with the appropriate sharing and security permissions.

About this task

Create or ensure that a domain or Windows account exists that all developers can use.

Parent topic: [Installing the developer workstation software components](#)

Importing encryption keys to Datacap computers

In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.

Before you begin

You must generate the encryption keys in the keystore on a server on which the Datacap server software component is installed. You export the new keys to a key transport file.

Procedure

To import encryption keys to Datacap computers:

1. Find the dc_KTF.xml key transport file in the c:\Datacap\Taskmaster folder on the Datacap server where you generated and exported the encryption keys.
2. Copy the dc_KTF.xml key transport file to the appropriate folder on the computer where you installed the new component. The encryption keys are automatically applied to the keystore the next time you start the Datacap component.

Table 1. Encryption key folder locations by component

Component	Folder
Datacap Server	C:\Datacap\Taskmaster
Datacap applications	C:\Datacap\Taskmaster
Datacap Desktop	C:\Datacap\DcDesktop
Datacap Studio	C:\Datacap\DStudio
Datacap FastDoc	C:\Datacap\FastDoc
Datacap Report Viewer	C:\Datacap\RV2\bin
Datacap Web Client	C:\Datacap\tmweb.net\bin
Datacap Web Services	C:\Datacap\wTM\bin

Parent topic: [Installing the developer workstation software components](#)

Setting up the scanner

When a computer is to be used as a scan station, follow the manufacturer's instructions to attach the scanner to the computer. Install and configure the required scanner drivers and software. Be sure you can scan successfully with an image capture software product other than Datacap. If you are unable to scan documents with software that is not Datacap, you cannot scan documents with Datacap.

About this task

Tip: A physical scanner is not required for running the Datacap that is separately licensed or sample applications because those applications can process the prescanned, sample images that are included during installation.

Parent topic: [Installing the developer workstation software components](#)

Configure Internet Explorer on the developer workstation

You must configure Internet Explorer on the developer workstation to enable access to Datacap Web Client, the system administration panel, and the user panel.

About this task

To set up the developer workstation, follow the steps in [Configuring and testing the remote workstation](#).

Parent topic: [Installing the developer workstation software components](#)

Copying the application to the Datacap Server

The following information provides instructions for copying specific applications. However, the instructions are equally applicable to a customized application, and you can refer to any one of the topics for copying your custom application. You need to start Datacap Studio, start the Datacap Studio Application Copy wizard, and copy the application from the developer workstation to the server. The application can then be accessed by other workstations that are running the Datacap Client or Datacap Web Client, and by other services and processes.

About this task

The Windows account that you use to log on must have full control for both sharing and security permissions on the Datacap folder on the Datacap Server. If the application includes custom Datacap web pages, your Windows account must have sharing permissions on the Datacap web server and for the Datacap folder. Your Windows account must also have full control for security permissions on the tmweb.net folder.

Important:

When you are copying a foundation or sample application to change it or customize it to your business needs, rename it. Instructions for renaming are included in [Copying the application to Datacap Server](#). Renaming the application achieves the following benefits:

- Avoid losing your customization when you are upgrading to later versions of Datacap. Application folders with the names of Datacap applications can be overwritten.
- Better identify the business function the application provides for your users
- Leave the foundation and sample applications intact in case you encounter issues with your customized version. You can use the original application for troubleshooting to determine whether or how the feature works in the original application as compared to your customized version
- Leave the foundation and sample applications intact so you can use them as models when you are building your own rulesets and tasks
- [Sharing the Datacap and tmweb.net folders on the Datacap web server](#)
When you want to copy a Datacap application that includes customized Datacap web pages, the Application wizard copies the pages to the Datacap web server. Before copying, you must share and set up the appropriate permissions for the C:\Datacap and C:\Datacap\tmweb.net folders on the web server.
- [Starting Datacap Studio to use the Application wizard](#)
To use the Application wizard to copy an application, start Datacap Studio without opening an application.
- [Setting or ensuring the correct Datacap.xml file is in use](#)
Ensure that Datacap Studio is using the correct version of the datacap.xml file. The correct version is the local file that contains an entry for the application that you want to copy from the developer workstation to the Datacap Server.
- [Copying the application to Datacap Server](#)
You can copy the application from the developer workstation to the Datacap Server by using the Datacap Studio Application Copy wizard.
- [Copying the Datacap.xml file from the developer workstation to the Datacap Server](#)
The datacap.xml file on the developer workstation contains information about the application that you copied and the location of the application's files on the Datacap Server.

- [Setting the location of the Datacap Server and the Datacap.xml file](#)
You must define the location of the datacap.xml and the location of the Datacap Server for the application that you want to run.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Sharing the Datacap and tmweb.net folders on the Datacap web server

When you want to copy a Datacap application that includes customized Datacap web pages, the Application wizard copies the pages to the Datacap web server. Before copying, you must share and set up the appropriate permissions for the C:\Datacap and C:\Datacap\tmweb.net folders on the web server.

About this task

After you finish copying the application, you can remove the permissions from the folders and stop sharing.

Procedure

1. On the web server, start Windows Explorer, browse to and right-click the C:\Datacap folder and select Properties.
2. Click the Sharing tab, then click Advanced Sharing. When User Account Control (UAC) is on, the User Account Control window is displayed. Click Yes.
3. Click Share this Folder and keep Datacap as the Share name.
4. Click Permissions.
5. Add or ensure that the domain/Windows user ID of the developer who is copying the application is set to allow Full Control.
6. Click OK to close the Permissions dialog.
7. Click OK to close the Advanced Sharing dialog.
8. Click Close to close the Properties dialog.
9. In Windows Explorer, browse to and right-click the C:\Datacap\tmweb.net folder and select Properties.
10. Click the Security tab, then click Edit.
11. Add or ensure that the domain/Windows user ID of the developer who is copying the application is set to allow Full Control.
12. Click OK to close the Permissions dialog.
13. Click Close to close the Properties dialog.

Parent topic: [Copying the application to the Datacap Server](#)

Starting Datacap Studio to use the Application wizard

To use the Application wizard to copy an application, start Datacap Studio without opening an application.

Procedure

To start Datacap Studio without opening an application:

1. On the developer workstation, in the Start menu, click IBM Datacap Developer Tools>Datacap Studio. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.
2. On the Applications window, click Close. An empty Datacap Studio main window opens on the Rulemanager tab.

Parent topic: [Copying the application to the Datacap Server](#)

Setting or ensuring the correct Datacap.xml file is in use

Ensure that Datacap Studio is using the correct version of the datacap.xml file. The correct version is the local file that contains an entry for the application that you want to copy from the developer workstation to the Datacap Server.

About this task

When you install the Datacap Client software on the developer workstation, the sampleDatacap applications are installed. A default version of the datacap.xml file is placed in the C:\Datacap folder. This datacap.xml file contains entries for the Datacap applications and allows Datacap Studio to locate the applications.

Procedure

To set the Datacap.xml file:

1. On the developer workstation, with an empty Datacap Studio main window open, click Settings.
2. Click the DCapp tab.
3. Set or ensure that the path in the Main application management file field is set to the local copy of the datacap.xml file. Then, click OK.
4. Close and restart Datacap Studio after you change the path to the datacap.xml file so you can use the new file.

Parent topic: [Copying the application to the Datacap Server](#)

Copying the application to Datacap Server

You can copy the application from the developer workstation to the Datacap Server by using the Datacap Studio Application Copy wizard.

About this task

Important: By its design, the Application Copy wizard does not copy batches or their associated records in the engine database to the server.

Procedure

1. On the developer workstation, with an empty Datacap Studio main window open, click Datacap application wizard. The Application Wizard Overview window opens.
2. Click Next. The Application Wizard Mode window opens.
3. Select the Copy an existing RRS application option, then click Next. The Application Wizard Copy an existing application window opens.
4. Enter or select information for the following fields:

Table 1. Entering information in the Application Wizard Copy an existing application window

Field	Description	Example
-------	-------------	---------

Field	Description	Example
Select an application to copy from the list	The items in this list are the applications that are listed in the datacap.xml file that is in use. If the application you are copying does not display, follow the instructions in Setting the Datacap.xml file to point to the correct datacap.xml file.	APT, MClaims, or TravelDocs
Root folder on target system	Target folder on the Server under which the new application folder is created.	\\Server\Datacap
Datacap Web folder	Location of the tmweb.net folder on the Datacap Web Client server. Leave blank if Datacap Web Client is not set up yet or if there are no custom web pages in the source application. If Datacap Web Client is set up and there are custom web pages in the source application, use the correct web server name and folder location of tmweb.net.	\\WebServer\Datacap\tmweb.net
Rename Copy	Indicates that you want the new application to have a different name than the original application.	Selected
New Name	Enter the name of the new application.	Datacap Accounts Payable example: Company Invoices Medical Claims example: Hospital

Field	Description	Example
		all Claims TravelDocs example: Employee Travel

You should rename an application when you are making a copy. When you provide a New Name, it cannot be a substring of the original application and it can be composed of only alphabetic, numeric, and underscore characters. For TravelDocs, the name cannot be longer than 50 characters.

5. Click Next. The Application Wizard Finish window opens.
6. Click Finish to start the copying process.

The wizard makes a copy of the application in the new location on the server. The wizard then runs search-and-replace operations throughout the application to make all changes. Messages and progress bars are displayed, and when the process is complete, the Application Wizard Summary window is displayed with one or more of the following messages:

- o Warnings: Indicate that the wizard encountered something unexpected, but might continue.
 - o General: Shows the results that completed successfully.
 - o Errors: Indicate that the wizard encountered a problem that must be corrected.
7. Click View Logs to open the appwiz.log file, found in the application folder. The application folders are located in the \\Server\Datacap directory on the server:

Datacap Accounts Payable:

 \APT

Medical Claims:

 \MClaims

TravelDocs:

 \TravelDocs

Search the log for the words `error`, `warning`, and `not copied` to locate the issues that you must address. The `appwiz.log` contains a list of the changes to the application made by the wizard:

Type of Change	Information Logged
Rename databases	Before and After values are displayed
Rename files within folders	Shows Before and After values are displayed
Update contents of databases	Messages are displayed that identify what was updated, Before and After values are displayed
Update strings within project files	Before and After values are displayed
Update parameters in rule files (.rul)	When changed: The Before and After values are displayed. When not changed: Information Messages are displayed - Depending on the message, you decide whether it must be changed, and if so, you change it manually.
Updates Fingerprint database name	Application.xml is displayed

8. Click Exit to close Datacap Studio.

Parent topic: [Copying the application to the Datacap Server](#)

Copying the Datacap.xml file from the developer workstation to the Datacap Server

The `datacap.xml` file on the developer workstation contains information about the application that you copied and the location of the application's files on the Datacap Server.

Procedure

1. On the developer workstation, start Windows Explorer, browse to, and open the `C:\Datacap` folder.
2. On the developer workstation, start another instance of Windows Explorer, browse to the Datacap Server, and open the `\\Server\Datacap` folder.
3. Copy the developer workstation's `datacap.xml` file and paste it into the `\\Server\Datacap` folder.

What to do next

After you copy the `datacap.xml` file to the Datacap Server, update it following the instructions in [Updating the datacap.xml file on the Datacap server](#).

Parent topic: [Copying the application to the Datacap Server](#)

Setting the location of the Datacap Server and the Datacap.xml file

You must define the location of the `datacap.xml` and the location of the Datacap Server for the application that you want to run.

Procedure

To define the location of the server and the datacap.xml file:

1. On the developer workstation, in the Start menu, click IBM Datacap Services>Datacap Application Manager. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.
2. Select the application to which you want to set the location, such as TravelDocs, or APT. The paths are displayed in the fields on the Main tab.
3. Ensure that all workflows are displayed. Check that all of the paths reflect the correct UNC paths to the various files and databases by using the Datacap Server name rather than C:\.
4. Click Locale. Select the option that is associated with the language and regional settings that are used on most of the documents to be processed by the application. If you do not select a locale, the value that is set on the Regional and Language property sheet of the operating system is used.
5. Click the Datacap tab and change the Name or IP address field to the IP address or the name of the Datacap Server without using backslashes.
6. In the Protocol field, select the TCP/IP version that is used on your network. If you do not select a protocol, Datacap defaults to TCP IPv4.
7. Click the Service tab and verify that the path reflects the correct UNC location of the datacap.xml on the server, such as \\Server\Datacap\datacap.xml.
8. Close Datacap Application Manager.

Parent topic: [Copying the application to the Datacap Server](#)

Complete the Datacap server configuration

You must update the datacap.xml file on the Datacap server to add references to applications. You must also set up security on the folder that contains the applications.

Complete the Datacap server configuration by updating the datacap.xml file on the Datacap server to reflect the locations of your applications. You must also set up security on the C:\Datacap\Application folder that contains your newly copied application.

- [Updating the datacap.xml file on the Datacap server](#)
The datacap.xml file on the Datacap server must contain references to the applications. The datacap.xml file must also contain the locations that you are configuring for use.
- [Setting up security on the Datacap\Application folder](#)
You must set up security for the shared C:\Datacap\Application folder by setting security permissions in the Datacap Properties Securities window.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Updating the datacap.xml file on the Datacap server

The datacap.xml file on the Datacap server must contain references to the applications. The datacap.xml file must also contain the locations that you are configuring for use.

About this task

Before you begin, identify the names and locations of the applications that you want the Datacap Application Service to be aware of.

Important: The Datacap Application Service is case-sensitive. When you add or change entries in the datacap.xml file, make sure that the case matches the case of the UNC paths, folders, and file names.

Procedure

1. Open the C:\Datacap\datacap.xml file in a text editor on the Datacap server. It contains lines that look like this example:

```
<datacap ver="8.0">
  <app name="Flex" ref="Flex"/>
  <app name="TravelDocs" ref="\\ServerName\Datacap\TravelDocs"/>
  <app name="APT" ref="APT"/>
  <app name="AppWizard" ref="DStudio\AppWizard"/>
  <app name="NENU" ref="NENU"/>
</datacap>
```

2. To delete references to applications that do not exist or are not configured, delete the corresponding `<app name=` lines. When the only application on the Datacap server is the TravelDocs application, delete all of the lines except the line for the TravelDocs application. The result looks like this example:

```
<datacap ver="8.0">
  <app name="TravelDocs" ref="\\ServerName\Datacap\TravelDocs"/>
</datacap>
```

3. To add references to applications that exist, add a line with the name of the application and the full UNC path to the application folder. When you add a line for the Maintenance Manager application, your result looks like this example:

```
<datacap ver="8.0">
  <app name="TravelDocs" ref="\\ServerName\Datacap\TravelDocs"/>
  <app name="NENU" ref="\\ServerName\Datacap\NENU"/>
</datacap>
```

4. Save and then close the datacap.xml file.

Parent topic: [Complete the Datacap server configuration](#)

Setting up security on the Datacap\Application folder

You must set up security for the shared C:\Datacap\Application folder by setting security permissions in the Datacap Properties Securities window.

About this task

Repeat these instructions for each C:\Datacap\Application folder on the server. By setting up these security permissions at this level, you grant access to different applications to specific developers and users.

Procedure

1. On the server, start Windows Explorer, navigate to, and right-click the C:\Datacap\Application folder and select Properties.
2. Click the Security tab, and then click Edit.
3. Add NETWORK SERVICE and local IUSR and set both to allow Full Control.
4. Ensure that the domain/Windows user ID of Datacap Web Client is set to allow Read & Execute.
5. Ensure that the domain/Windows user ID of developer is set to allow Full Control.
6. Ensure that the domain/Windows user ID of Datacap server service is set to allow Full Control.
7. When the batches folders are staying on Server in the C:\Datacap\Application path, add or ensure that the domain/Windows user IDs of Datacap users are set to allow Full Control.

Parent topic: [Complete the Datacap server configuration](#)

Complete the Datacap Web Client server configuration

To complete the Datacap Web Client server configuration, you must set the location of the Datacap.xml file. Then, you must restart Internet Information Services.

- [Setting the location of the datacap.xml file](#)
You must define the location of the Datacap application settings by using the Datacap Application Manager. The datacap.xml file contains the application settings.
- [Restarting Internet Information Services \(IIS\)](#)
After you set the location of the datacap.xml file on a web server, you must restart Internet Information Services (IIS).

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Setting the location of the datacap.xml file

You must define the location of the Datacap application settings by using the Datacap Application Manager. The datacap.xml file contains the application settings.

About this task

The datacap.xml file that contains the application settings is on the Datacap server.

Procedure

To set the location of the datacap.xml file:

1. In the Start menu, select IBM Datacap Services>Datacap Application Manager. If the User Account Control window opens, click Yes.
2. Click the Service tab to display it.
3. Change or ensure that the path reflects the correct location of the datacap.xml file. For example, \\Server\Datacap\datacap.xml
4. Close the Datacap Application Manager.

Parent topic: [Complete the Datacap Web Client server configuration](#)

Restarting Internet Information Services (IIS)

After you set the location of the datacap.xml file on a web server, you must restart Internet Information Services (IIS).

Parent topic: [Complete the Datacap Web Client server configuration](#)

Configuring and testing the remote workstation

You can configure and test a remote workstation that uses Internet Explorer to access Datacap Web Client.

About this task

This section provides the instructions you need to configure a remote workstation that uses Internet Explorer to access Datacap Web Client. You need to know the IP address or the server name of your Datacap Web

Server before you can configure a remote workstation.

Procedure

Determine the way that you want to configure the remote workstation. There are two ways to configure a remote workstation:

Option	Description
You can package the Datacap Web Client Configuration Tool, send the package to the user, and have the user run the tool to configure Internet Explorer.	To package the Datacap Web Client Configuration Tool, follow the instructions in Packaging the Datacap Web Client Configuration Tool .
The user can configure Internet Explorer manually. Provide the user with these instructions.	Configuring Internet Explorer manually Test Internet Explorer

- [Packaging the Datacap Web Client Configuration Tool](#)
You can create a package for a user to configure Internet Explorer on a remote workstation to access Datacap Web Client.
- [Adding the TMWeb.net address as a trusted site](#)
You must add the TMWeb.net address as a trusted site in a client/server configuration to prevent Internet Explorer from blocking access to the Datacap Web Client site.
- [Configuring and testing Internet Explorer using the Web Client Configuration package](#)
To ensure remote access with Datacap Web Client, you must enable the Internet Explorer security option and set the configuration in the Datacap Web Client Configuration.
- [Configuring Internet Explorer manually](#)
You must configure Internet Explorer on each workstation with the Datacap Web Client.
- [Testing Internet Explorer](#)
You can test your Datacap Web Client configuration with the Internet Explorer test page.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Packaging the Datacap Web Client Configuration Tool

You can create a package for a user to configure Internet Explorer on a remote workstation to access Datacap Web Client.

About this task

Package the Datacap Web Client Configuration Tool for a user to configure Internet Explorer to access the Datacap Web Client.

Procedure

1. On any machine where Datacap software is installed, open Windows Explorer, go to the C:\Datacap\support\WebConfiguration folder and open the folder.
2. Make a copy of the WebClientConfig.exe.config file.
3. With a text editor such as Notepad, open the WebClientConfig.exe.config file.
4. Locate the following line.

```
<value>http://localhost/tmweb.net</value>
```

5. Change localhost to the IP address or server name of your Datacap Web Client server.
6. Save and close the WebClientConfig.exe.config file.

7. Send the following three files to your user: WebClientConfig.exe.config, Datacap.Config.dll, and WebClientConfig.exe. Also send the URL for the Datacap Web Client Internet Explorer test page, such as <http://WebServerName/tmweb.net/ietest.aspx>.

Parent topic: [Configuring and testing the remote workstation](#)

Adding the TMWeb.net address as a trusted site

You must add the TMWeb.net address as a trusted site in a client/server configuration to prevent Internet Explorer from blocking access to the Datacap Web Client site.

Procedure

1. On the computer on which you want to use Internet Explorer to access Datacap Web Client, start the 32-bit version of Internet Explorer.
2. On the Tools menu, select Internet Options. The Internet Options dialog opens.
3. Click the Security tab.
4. Select Trusted sites, then click Sites. The Trusted Sites dialog opens.
5. Clear the Require server verification option when your Datacap Web Client URL begins with http://.
6. Depending on whether you are running Internet Explorer on the same machine as Datacap Web Client, do one of the following:
 - o When you are running Internet Explorer on the same machine, enter the default server address (<http://localhost>) in the Add this website to the zone field, then click Add.
 - o When you are running Internet Explorer from a different machine, enter either the web server IP address or name (<http://WebServerName>) in the Add this website to the zone field, then click Add.

Parent topic: [Configuring and testing the remote workstation](#)

Configuring and testing Internet Explorer using the Web Client Configuration package

To ensure remote access with Datacap Web Client, you must enable the Internet Explorer security option and set the configuration in the Datacap Web Client Configuration.

Before you begin

Before you can begin, you must have or do all of the following:

- A 32-bit version of Internet Explorer must be installed and running on your workstation.
- You must have the instructions for Adding the TMWeb.net address as a trusted site.
- You must have the complete URL for the Internet Explorer test page on your Datacap Web Client remote access site, for example <http://WebServerName/tmweb.net/ietest.aspx>.
- You must have three files from your Administrator named WebClientConfig.exe.config, Datacap.config.dll, and WebClientConfig.exe.

Important: The login user must be a member of the local administrator group for WebClientConfig.exe to work properly. If the login user is not a member of the local administrator group, then disable UAC, temporarily add the login user to the local administrator group, and after configuration, remove the login user from the local administrator group.

About this task

Datacap Web Client can be accessed from a remote workstation using the 32-bit version of Internet Explorer. This procedure provides instructions on how to configure Internet Explorer and test your access to Datacap Web Client using the Web Client Configuration package provided to you by your Administrator.

Procedure

1. Start Internet Explorer on your workstation and ensure you can view web pages on the Internet.
2. Follow the instructions to add the URL for Datacap Web Client as a trusted site.
3. On the Tools menu, select Internet Options. The Internet Options dialog opens.
4. Click the Security tab, select Trusted sites, then click the Custom level button.
5. Scroll down until the Miscellaneous settings are displayed.
6. The Include local directory path when uploading files to a server option should be Enabled.
7. Click OK, click Yes if a warning is displayed, then click OK to close the Internet Options dialog.
8. Close Internet Explorer.
9. Place the three files on your workstation either in a folder on your desktop or in the location that is recommended by your Administrator.
10. Double-click the WebClientConfig.exe file. When UAC is on, the User Access Control window opens. Click Yes. The Datacap Web Client Configuration window opens and displays a URL.
11. Ensure that the web server name that is displayed matches the web server information (server name or IP address) that you added to Internet Explorer as a trusted site. If it does not match, contact your Administrator for the correct URL and enter it manually.
12. Click Configure, then click OK to close the message box that indicates the configuration was successful.
13. Click Exit.
14. Start Internet Explorer and enter the URL to the Internet Explorer test page, such as `http://WebServerName/tmweb.net/ietest.aspx` then press Enter. When UAC is on, the first time you access the test page, the User Access Control window opens. Click Yes each time you are prompted. Clicking Yes downloads the required Datacap TIFF Viewer, Thumbnails, and DataEdit Controls. After the downloads are complete, the IE Test Page is displayed.
15. Click Test. The red Xs change to green checkmarks when the test completes successfully. If the TIFF viewer test does not pass, switch to the 32-bit version of Internet Explorer and repeat this procedure from step 1.

Parent topic: [Configuring and testing the remote workstation](#)

Configuring Internet Explorer manually

You must configure Internet Explorer on each workstation with the Datacap Web Client.

Procedure

To configure Internet Explorer manually:

1. Start Internet Explorer and ensure you can view web pages on the Internet.
2. On the Tools menu, select Internet Options. The Internet Options dialog opens.
3. Click the Security tab.
4. Select Trusted Sites, then click Sites. The Trusted Sites dialog opens.
5. Clear the Require server verification option when your Datacap Web Client URL begins with `http://`.
6. When you are running Internet Explorer on the same machine as Datacap Web Client, enter the default web server address (`http://localhost`) in the Add this website to the zone field, then click Add. The URL is added to the pane. When you are running Internet Explorer from a different machine, enter either the web server IP address or name (`http://WebServerName`) in the Add this website to the zone field, then click Add. The URL is added to the pane.
7. Click Close. The Internet Options dialog is displayed.

8. On the Security tab, click the Custom level button. The Security Settings dialog appears.
9. Scroll down to the ActiveX controls and plug-ins settings.
10. Set the following options to Enabled.
 - o Download signed ActiveX controls
 - o Initialize and script ActiveX controls not marked as safe for scripting
11. Scroll down to the Miscellaneous settings.
12. Set the Include local directory path when uploading files to a server option to Enabled.
13. Click OK to confirm your changes.

Parent topic: [Configuring and testing the remote workstation](#)

Testing Internet Explorer

You can test your Datacap Web Client configuration with the Internet Explorer test page.

About this task

This procedure provides instructions on how to test your configuration of Internet Explorer with Datacap Web Client.

Procedure

1. Start Internet Explorer.
2. When you are running Internet Explorer on the same machine as Datacap Web Client, enter the URL for Datacap Web Client (<http://localhost>) followed by the `tmweb.net` virtual directory and the test page, <http://localhost/tmweb.net/ietest.aspx>. When you are running Internet Explorer from a different machine, enter either the web server address or name, followed by the `tmweb.net` virtual directory and the test page, such as `http://WebServerName/tmweb.net/ietest.aspx`. When UAC is on, the first time you access the test page, the User Access Control window opens. Click Yes to download the Datacap TIFF Viewer, Thumbnails, and DataEdit Controls. The Internet Explorer test page is displayed.
3. Click Test. The red Xs change to green check marks when the test completes successfully. If the Datacap TIFF Viewer does not download, switch to the 32-bit version of Internet Explorer and repeat this procedure from step 1.

Parent topic: [Configuring and testing the remote workstation](#)

Installing the Datacap client on the user workstation

You can run the Datacap installation program wizard on a user workstation to install the necessary Datacap client software.

About this task

This procedure provides instructions on how to run the Datacap installation program wizard on an user workstation to install the necessary Datacap software components. The components that you install include the Datacap client and the separately licensed connectors for which you have a license.

Procedure

1. Make the installation package available on your network or insert the Datacap CD in the Workstation's CD/DVD drive. If the installation process does not start automatically, or if the package is on the

- network, open Windows Explorer, navigate to and double-click the Setup.exe. When User Access Control (UAC) is on, the User Account Control window opens. Click Yes.
2. Select the appropriate language, then click OK. The language that you select determines the language that is displayed by the installation program wizard during the installation process.
 3. When additional, redistributable software is required, the installation program wizard displays a list of the items that must be installed. Click Install.
 4. Click Next.
 5. Accept the license agreement. Then, click Next.
 6. Select the Custom option. Then, click Next.
 7. Exclude all components except Datacap client and the separately licensed connectors for which you have a license.
 8. Expand the Datacap client software component and exclude the Applications, Datacap Studio, and Maintenance Manager components. If you are not using the FastDoc user interface, exclude that as well.
 9. Click Next.
 10. Click Install.
 11. Click Finish.

- [Ensure that accounts exist for users](#)

Ensure that a domain/Windows account exists for each Datacap user. Datacap does not require that a unique Windows account is set up for each user. A user can use any Windows account that is set up with the appropriate sharing and security permissions.

- [Importing encryption keys to Datacap computers](#)

In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.

- [Setting up sharing and security permissions for users](#)

You must set up sharing and security permissions to Datacap folders for each Datacap user.

- [Setting up the scanner](#)

When a computer is to be used as a scan station, follow the manufacturer's instructions to attach the scanner to the computer. Install and configure the required scanner drivers and software. Be sure you can scan successfully with an image capture software product other than Datacap. If you are unable to scan documents with software that is not Datacap, you cannot scan documents with Datacap.

- [Setting the location of the datacap.xml file](#)

You must define the location of the Datacap application settings by using the Datacap Application Manager. The datacap.xml file contains the application settings.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Ensure that accounts exist for users

Ensure that a domain/Windows account exists for each Datacap user. Datacap does not require that a unique Windows account is set up for each user. A user can use any Windows account that is set up with the appropriate sharing and security permissions.

Parent topic: [Installing the Datacap client on the user workstation](#)

Importing encryption keys to Datacap computers

In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.

Before you begin

You must generate the encryption keys in the keystore on a server on which the Datacap server software component is installed. You export the new keys to a key transport file.

Procedure

To import encryption keys to Datacap computers:

1. Find the dc_KTF.xml key transport file in the c:\Datacap\Taskmaster folder on the Datacap server where you generated and exported the encryption keys.
2. Copy the dc_KTF.xml key transport file to the appropriate folder on the computer where you installed the new component. The encryption keys are automatically applied to the keystore the next time you start the Datacap component.

Table 1. Encryption key folder locations by component

Component	Folder
Datacap Server	C:\Datacap\Taskmaster
Datacap applications	C:\Datacap\Taskmaster
Datacap Desktop	C:\Datacap\DcDesktop
Datacap Studio	C:\Datacap\DStudio
Datacap FastDoc	C:\Datacap\FastDoc
Datacap Report Viewer	C:\Datacap\RV2\bin
Datacap Web Client	C:\Datacap\tmweb.net\bin
Datacap Web Services	C:\Datacap\wTM\bin

Parent topic: [Installing the Datacap client on the user workstation](#)

Setting up sharing and security permissions for users

You must set up sharing and security permissions to Datacap folders for each Datacap user.

About this task

This procedure provides instructions on how to grant the appropriate permissions to Datacap user accounts.

Procedure

1. Set up sharing permissions on the \Datacap folder.
 - a. On the Server, start Windows Explorer, navigate to and right-click the C:\Datacap folder and select Properties.
 - b. Click the Sharing tab and then click Advanced Sharing.
 - c. Click Permissions.
 - d. When the batches folders are staying on server in C:\Datacap\Application path, add or ensure that the domain/Windows user IDs of Datacap users are set to allow Full Control.
2. Set up security on the \Datacap folder.
 - a. On the server, start Windows Explorer, navigate to, and right-click the C:\Datacap folder and select Properties.

- b. Click the Security tab.
 - c. When the batches folders are staying on server in C:\Datacap\Application path, add or ensure that the domain/Windows user IDs of Datacap users are set to allow Read & Execute.
3. Set up security on the \Datacap\Application folder.
 - a. On the server, start Windows Explorer, navigate to, and right-click the C:\Datacap\Application folder and select Properties.
 - b. Click the Security tab and click Edit.
 - c. When the batches folders are staying on server in the C:\Datacap\Application path, add or ensure that the domain/Windows user IDs of Datacap users are set to allow Full Control.

Parent topic: [Installing the Datacap client on the user workstation](#)

Setting up the scanner

When a computer is to be used as a scan station, follow the manufacturer's instructions to attach the scanner to the computer. Install and configure the required scanner drivers and software. Be sure you can scan successfully with an image capture software product other than Datacap. If you are unable to scan documents with software that is not Datacap, you cannot scan documents with Datacap.

About this task

Tip: A physical scanner is not required for running the Datacap that is separately licensed or sample applications because those applications can process the prescanned, sample images that are included during installation.

Parent topic: [Installing the Datacap client on the user workstation](#)

Setting the location of the datacap.xml file

You must define the location of the Datacap application settings by using the Datacap Application Manager. The datacap.xml file contains the application settings.

About this task

The datacap.xml file that contains the application settings is on the Datacap server.

Procedure

To set the location of the datacap.xml file:

1. In the Start menu, select IBM Datacap Services>Datacap Application Manager. If the User Account Control window opens, click Yes.
2. Click the Service tab to display it.
3. Change or ensure that the path reflects the correct location of the datacap.xml file. For example, \\Server\Datacap\datacap.xml
4. Close the Datacap Application Manager.

Parent topic: [Installing the Datacap client on the user workstation](#)

Installing and configuring Datacap Report Viewer

You can install and configure Report Viewer to run locally or in a Datacap client/server environment so you can use the browser-based Datacap Web Client for processing.

- [Datacap Report Viewer](#)
The Datacap Report Viewer web application displays real-time reports of activity that is related to your Datacap applications.
- [Client/server environment: Overview of Datacap Report Viewer installation and configuration](#)
You must complete the required tasks to install and configure the Datacap Report Viewer reporting component.
- [Creating or ensuring an account exists for Report Viewer](#)
If a domain/Windows account does not exist for Report Viewer, you must create one.
- [Installing and configuring Datacap Report Viewer on a web server](#)
You can install and configure Report Viewer on a Windows 2008 web server.
- [Adding the Datacap Report Viewer address as a trusted site](#)
Add the Report Viewer web address as a trusted site to prevent Internet Explorer from blocking access to the site.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Datacap Report Viewer

The Datacap Report Viewer web application displays real-time reports of activity that is related to your Datacap applications.

For a list of the standard reports, see [Standard reports](#).

Parent topic: [Installing and configuring Datacap Report Viewer](#)

Client/server environment: Overview of Datacap Report Viewer installation and configuration

You must complete the required tasks to install and configure the Datacap Report Viewer reporting component.

Before you start, ensure that you met the installation and configuration prerequisites.

In a client/server environment:

1. Create or ensure that an account exists for Report Viewer.
2. Ensure that you have the required Microsoft Internet Information Services (IIS) components installed.
3. Set up various Report Viewer permissions.
4. Install Report Viewer on the web server.
5. Import encryption keys on the web server.
6. Add an Application Pool for Report Viewer.
7. Set up the Report Viewer website and Application Pool advanced settings.

Parent topic: [Installing and configuring Datacap Report Viewer](#)

Creating or ensuring an account exists for Report Viewer

If a domain/Windows account does not exist for Report Viewer, you must create one.

About this task

Datacap does not require you to set up a unique Windows account for Report Viewer. Report Viewer can use any Windows account if an account can be set up with the appropriate sharing and security permissions.

Important: When Report Viewer and Datacap Web Client are installed on the same web server, they must use the same domain/Windows account.

Parent topic: [Installing and configuring Datacap Report Viewer](#)

Installing and configuring Datacap Report Viewer on a web server

You can install and configure Report Viewer on a Windows 2008 web server.

Before you begin

Before installing Datacap Report Viewer, [stop or ensure the Datacap Server Service is stopped](#).

- [Verifying that IIS components are installed](#)
Ensure that the IIS Web Server Role Services are installed for the Datacap Web Client server components, such as Datacap Web Client, Report Viewer, Fingerprint Service, and Datacap Web Services.
- [Setting up sharing permissions for Datacap Report Viewer on the Datacap folder](#)
You must set up the appropriate sharing permissions for the Report Viewer account on the Datacap Server shared C:\Datacap folder.
- [Setting up security for Datacap Report Viewer on the Datacap folder](#)
You must set up the appropriate security permissions on the C:\Datacap folder on the server when the Datacap Server operating system is Windows 2008. Other accounts were granted security permissions during the installation and configuration of Datacap.
- [Installing Datacap Report Viewer on the web server](#)
Run the Datacap installation wizard on a web server to install the Datacap Report Viewer software component.
- [Importing encryption keys to Datacap computers](#)
In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.
- [Adding an application pool for Report Viewer](#)
When you are running on Windows 7 or Windows 2008, you must add a Microsoft Internet Information Services (IIS) Application Pool for use by Report Viewer.
- [Client/server environment: Setting up the Datacap Report Viewer website](#)
You must set up the Report Viewer website on Microsoft Internet Information Services (IIS) 7.5.
- [Setting the location of the datacap.xml file](#)
You must define the location of the Datacap application settings by using the Datacap Application Manager. The datacap.xml file contains the application settings.

Parent topic: [Installing and configuring Datacap Report Viewer](#)

Verifying that IIS components are installed

Ensure that the IIS Web Server Role Services are installed for the Datacap Web Client server components, such as Datacap Web Client, Report Viewer, Fingerprint Service, and Datacap Web Services.

About this task

The Datacap Web Client server components must have specific IIS Web Server Role Services installed.

Procedure

Ensure that the IIS components are installed.

1. From the Start menu on the web server, select Administrative Tools > Server Manager.
2. In the Server Manager hierarchy pane, expand Roles and select Web Server (IIS).
3. In the Web Server (IIS) pane, expand Role Services. Under Common HTTP Features ensure that Static Content, Default Document, Directory Browsing, and HTTP Errors are installed. If you are installing wTM, do not install the WebDAV Publishing role service because it prevents the Datacap web services PUT method from functioning.
4. In the Role Services pane under Application Development, ensure that the ASP.NET, .NET Extensibility, ASP, ISAPI Extensions, and ISAPI Filters are installed.
5. In the Role Services pane, under Health and Diagnostics ensure that HTTP Logging and Request Monitor are installed.
6. In the Role Services pane, under Security ensure that Request Filtering is installed.
7. In the Role Services pane, under Performance ensure that Static Content Compression is installed.
8. In the Role Services pane, under Management Tools ensure that IIS Management Console is installed.
9. (Windows Server 2012 R2) In Server Manager, click Dashboard, and click Add roles and features.
 - a. In the Add Roles and Features Wizard, go to the Features section.
 - b. In .NET Framework 4.5 Features > WCF Services, select the HTTP Activation check box.
10. Close the Server Manager window.

Parent topic: [Installing and configuring Datacap Report Viewer on a web server](#)

Setting up sharing permissions for Datacap Report Viewer on the Datacap folder

You must set up the appropriate sharing permissions for the Report Viewer account on the Datacap Server shared C:\Datacap folder.

About this task

These instructions apply when the operating system of the Datacap Server is Windows 2008. Other accounts were already granted sharing permissions during the installation and configuration of Datacap.

Procedure

To set up sharing permissions for Report Viewer on the Datacap folder:

1. On the Datacap Server, start Windows Explorer, go to and right-click the C:\Datacap folder and select Properties.
2. Click the Sharing tab and click Advanced Sharing. If the User Account Control window opens, click Yes.
3. Click Permissions and ensure the domain/Windows user ID of Report Viewer is set to allow Read.

Parent topic: [Installing and configuring Datacap Report Viewer on a web server](#)

Setting up security for Datacap Report Viewer on the Datacap folder

You must set up the appropriate security permissions on the C:\Datacap folder on the server when the Datacap Server operating system is Windows 2008. Other accounts were granted security permissions during the

installation and configuration of Datacap.

Procedure

To set up security for Report Viewer on the Datacap folder:

1. On the Server, start Windows Explorer, go to and right-click the C:\Datacap folder and select Properties.
2. Click the Security tab and click Edit. If the User Account Control window opens, click Yes.
3. Add or ensure that the domain/Windows account for Report Viewer is set to allow Read & Execute.

Parent topic: [Installing and configuring Datacap Report Viewer on a web server](#)

Installing Datacap Report Viewer on the web server

Run the Datacap installation wizard on a web server to install the Datacap Report Viewer software component.

About this task

These instructions apply to servers that are running on Windows 2008.

Procedure

To install Report Viewer on the web server, follow this procedure.

1. Put the installation package on your network, or insert the Datacap CD in the CD/DVD drive on the server. If the installation process does not start automatically, or if the package is on the network, open Windows Explorer, go to and double-click the Setup.exe. If the User Account Control window opens, click Yes.
 2. Select the appropriate language and click OK. The language that you select determines the language that is displayed by the installation program during the installation process.
 3. When the installation requires other software, the installation program displays a list of the items to be installed. Click Install.
 4. Click Next.
 5. Accept the license agreement and click Next.
 6. Select the Custom option and click Next.
 7. Exclude all components from the installation process except Report Viewer.
 8. Click Next.
 9. Click Install.
 10. Click Finish.
- [Enabling ADSI or LDAP authentication with Report Viewer](#)
In addition to any configuration on the Datacap Server, Report Viewer has its own requirement for using ADSI or LDAP authentication. Report Viewer includes an EnableLDAP setting that is used for ADSI and LDAP authentication.

Parent topic: [Installing and configuring Datacap Report Viewer on a web server](#)

Enabling ADSI or LDAP authentication with Report Viewer

In addition to any configuration on the Datacap Server, Report Viewer has its own requirement for using ADSI or LDAP authentication. Report Viewer includes an EnableLDAP setting that is used for ADSI and LDAP authentication.

About this task

If the default EnableLDAP value of `false` is unchanged, then the login to Report Viewer requires the operator to enter a non-blank password.

Procedure

To change the default EnableLDAP value:

1. Confirm that no operators are using Report Viewer.
2. Log in to the Report Viewer web server.
3. Make a backup copy of the `\Datacap\RV2\web.config` file.
4. Open `web.config` in a text editor and locate the following line.

```
<add key="EnableLDAP" value="false"/>
```

5. Change `false` to `true`:

```
<add key="EnableLDAP" value="true"/>
```

6. Save the `web.config` file.

Parent topic: [Installing Datacap Report Viewer on the web server](#)

Importing encryption keys to Datacap computers

In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.

Before you begin

You must generate the encryption keys in the keystore on a server on which the Datacap server software component is installed. You export the new keys to a key transport file.

Procedure

To import encryption keys to Datacap computers:

1. Find the `dc_KTF.xml` key transport file in the `c:\Datacap\Taskmaster` folder on the Datacap server where you generated and exported the encryption keys.
2. Copy the `dc_KTF.xml` key transport file to the appropriate folder on the computer where you installed the new component. The encryption keys are automatically applied to the keystore the next time you start the Datacap component.

Table 1. Encryption key folder locations by component

Component	Folder
Datacap Server	C:\Datacap\Taskmaster
Datacap applications	C:\Datacap\Taskmaster
Datacap Desktop	C:\Datacap\DcDesktop
Datacap Studio	C:\Datacap\DStudio

Component	Folder
Datacap FastDoc	C:\Datacap\FastDoc
Datacap Report Viewer	C:\Datacap\RV2\bin
Datacap Web Client	C:\Datacap\tmweb.net\bin
Datacap Web Services	C:\Datacap\wTM\bin

Parent topic: [Installing and configuring Datacap Report Viewer on a web server](#)

Adding an application pool for Report Viewer

When you are running on Windows 7 or Windows 2008, you must add a Microsoft Internet Information Services (IIS) Application Pool for use by Report Viewer.

Procedure

To add an Application Pool for Report Viewer:

1. From the Windows Start menu, select Administrative Tools > Internet Information Services (IIS) Manager.
2. In the Connections pane, expand the computer, right-click Application Pools and select Add Application Pool.
3. Set the Name to `Report Viewer`.
4. Set the .NET Framework version to `.NET Framework v4.0.30319`.
5. Set the Managed pipeline mode to `Integrated`.
6. Select the Start application pool immediately option and click OK.

Parent topic: [Installing and configuring Datacap Report Viewer on a web server](#)

Client/server environment: Setting up the Datacap Report Viewer website

You must set up the Report Viewer website on Microsoft Internet Information Services (IIS) 7.5.

Procedure

To set up the Report Viewer website:

1. From the Windows Start menu on the web server, select Administrative Tools > Internet Information Services (IIS) Manager.
2. In the Connections pane, expand the computer and expand Sites. Then, right-click the Default Web Site and select Add Application.
3. Set the Alias to `Report Viewer`.
4. Click Select and select the Report Viewer Application Pool that you added, then click OK.
5. Set the Physical path by entering or browsing to the Report Viewer installation folder. The default location is `C:\Datacap\RV2`.
6. Click OK to close the Add Application dialog.
7. In the Connections pane, select Application Pools.
8. In the Application Pools pane, select the Report Viewer application pool.
9. In the Actions pane, in the Edit Application Pool section, click Advanced Settings.

10. Ensure that the Microsoft .NET version is set to v4.0.
11. Ensure that Enable 32-Bit Applications is set to True.
12. In the Process Model section, click Browse next to Identity.
13. In the Application Pool Identity window, select Custom account and click Set.
14. In the Set Credentials window, enter the Report Viewer domain/Windows account information (the same account that you added to the web server Administrators Group) in the format `accountname@domainname`. Then, enter the account password twice and click OK.
15. In the Process Model section, set Load User Profile to True.
16. Click OK.
17. In the Connections pane, expand the computer and expand Sites. Then, expand the Default Web Site and select the Report Viewer site, and in the middle pane, double-click Session State.
18. Under Cookie Settings, change the Name to `Report Viewer` or another unique name, then, in the Actions pane, click Apply.
19. In the Connections pane, select the Default Web Site, then, in the Actions pane, under Manage Web Site, click Restart.
20. Confirm that the Web Server, Application Pool, and Default website are started.

Parent topic: [Installing and configuring Datacap Report Viewer on a web server](#)

Setting the location of the datacap.xml file

You must define the location of the Datacap application settings by using the Datacap Application Manager. The datacap.xml file contains the application settings.

About this task

The datacap.xml file that contains the application settings is on the Datacap server.

Procedure

To set the location of the datacap.xml file:

1. In the Start menu, select IBM Datacap Services>Datacap Application Manager. If the User Account Control window opens, click Yes.
2. Click the Service tab to display it.
3. Change or ensure that the path reflects the correct location of the datacap.xml file. For example, `\\Server\Datacap\datacap.xml`
4. Close the Datacap Application Manager.

Parent topic: [Installing and configuring Datacap Report Viewer on a web server](#)

Adding the Datacap Report Viewer address as a trusted site

Add the Report Viewer web address as a trusted site to prevent Internet Explorer from blocking access to the site.

Procedure

To add the Report Viewer address as a trusted site:

1. Open Internet Explorer.
2. On the Tools menu, select Internet Options.
3. Click the Security tab.

4. Select Trusted sites, then click Sites.
5. Clear the Require server verification option when your Datacap Web URL begins with http://.
6. Depending on whether you are running Internet Explorer on the same computer as Report Viewer, do one or both of the following steps.
 - o Enter the default server address (http://127.0.0.1) in the Add this website to the zone field and click Add.
 - o Enter the web server IP address or name (http://WebServerName) in the Add this website to the zone field and click Add.
7. Click Close.
8. Click OK.

Parent topic: [Installing and configuring Datacap Report Viewer](#)

Installing and configuring the Rulerunner Service

The Rulerunner Service is a Datacap component that runs Datacap Studio application tasks in the background.

Background tasks are run without any human interaction. Background tasks are typically vScan, recognition, image pre-processing, validation, and export tasks.

Rulerunner supports simultaneous processes running on a single multiprocessor or multi-core equipped computer.

The ability to configure Rulerunner to run a single process is included as part of the standard Datacap licensing. Configuring Rulerunner to run multiple processes requires additional licensing.

With Rulerunner you can:

- Complete your document processing work faster. Faster turn-around of documents such as claims, contracts, tax returns and invoices improves responsiveness and shortens capture cycle times.
- When Rulerunner runs multiple processes, more of the physical resources of each machine are used which can reduce the total number of servers dedicated to the capture process.
- [Rulerunner configuration assumptions](#)
You must prepare your environment before you can install, configure, and run Datacap background tasks by using Rulerunner.
- [Overview of Rulerunner installation in a client/server environment](#)
You can install and configure the Rulerunner Service to run on one or more Rulerunner servers that are separate from the Datacap Server.
- [Configuring Rulerunner authentication](#)
When you install Rulerunner on one or more servers to run background Datacap tasks, you must authenticate Rulerunner as a user.
- [Installing and configuring the Rulerunner Service](#)
You can install and configure the Rulerunner Service in an environment where Datacap is running successfully. Where the Workstation and Server are separate machines. And where you are adding one or more additional Rulerunner servers to host instances of the Rulerunner Service.
- [Configuring Rulerunner to run your applications](#)
You can configure one or more instances of Rulerunner to run background tasks from your applications or workflows.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Rulerunner configuration assumptions

You must prepare your environment before you can install, configure, and run Datacap background tasks by using Rulerunner.

The instructions assume that all of the applications that you want Rulerunner to process have already been installed, configured, and run successfully using the Datacap client software. The instructions also assume you have one or more separate servers (Rulerunner server) on which you will install the Rulerunner Service.

Before you begin installing Rulerunner, ensure:

- You have a working Datacap 9.0.1 or later client/server environment.
- At a minimum, ensure that Datacap runs correctly in a client/server environment where the application files are located on the Datacap Server and the application runs correctly from a Workstation. This makes troubleshooting easier, and narrows down the sources of potential problems.
- If you want Datacap Rulerunner to run tasks from two or more Datacap applications, ensure that those applications are working and that you have manually processed batches from start to finish successfully.
- When installing on client and server computers, ensure that you have administrator access to all computers on which Datacap software is installed and to the servers running Windows Server 2008 on which you will install the Rulerunner Service.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Overview of Rulerunner installation in a client/server environment

You can install and configure the Rulerunner Service to run on one or more Rulerunner servers that are separate from the Datacap Server.

You can also configure Rulerunner to run tasks from a single or multiple Datacap applications on multiple threads.

Depending on how your Datacap environment does authentication, you must authenticate Rulerunner the same way. Instructions are provided for authenticating Rulerunner by using either Datacap authentication or an external authentication system (ADSI, LDAP, ADLDS, LLDAP).

Depending on your overall Datacap configuration, you might need to repeat one or more sets of instructions:

- Multiple Rulerunner servers - When you are installing the Rulerunner Service on more than one server, repeat the instructions for each Rulerunner server in the section titled [Installing and configuring the Rulerunner Service](#).
- Multiple Datacap applications - When you are configuring the Rulerunner Service to run tasks from more than one application, repeat the instructions for setting up an application in the section titled [Configuring Rulerunner to run your applications](#).
- Multiple workflows in a single application - When you are configuring the Rulerunner Service to run tasks from multiple workflows in a single application, configure each workflow as if it is its own application. You can refer to the instructions for setting up an application in the section titled [Configuring Rulerunner to run your applications](#).

The installation of Rulerunner in a client/server environment requires the following steps:

1. Ensure that you meet the prerequisites that are detailed in *Installation and Configuration Prerequisites*.
2. Ensure that you can manually run all application tasks in all applications before you begin this installation process.
3. Determine which Datacap tasks from which applications you want Rulerunner to process.

4. Regardless of whether you are using the Datacap authentication or an external authentication system, create or ensure that a domain/Windows account exists for Rulerunner. All instances of Rulerunner can use the same domain/Windows account. Follow the setup steps for your Datacap applications that are relevant to the type of authentication in use.
5. Configure Rulerunner permissions on the Datacap Server.
6. Install the Rulerunner Service on each Rulerunner server.
7. Install all of the Datacap software components that are required by the tasks that Rulerunner runs on each Rulerunner server.
8. On each Rulerunner server, set the location of the datacap.xml file on the Datacap Server.
9. Update the Rulerunner tab in the Application Manager with the appropriate Task profile names.
10. Complete the Rulerunner Service authentication setup.
11. Configure the Rulerunner Service account and permissions on each Rulerunner server.

Configuring the Rulerunner Service to process specific application tasks involves the following steps:

1. Configure the Rulerunner Service to run the appropriate applications and tasks.
2. Configure the security permissions for the Rulerunner Service on the various folders.
3. Create batches with the Datacap client.
4. Start the Rulerunner Service on the Rulerunner server.
5. Monitor the batches that the Rulerunner Service is processing.
6. Restart previously running Datacap software.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Related concepts:

[Installation and configuration prerequisites](#)

Configuring Rulerunner authentication

When you install Rulerunner on one or more servers to run background Datacap tasks, you must authenticate Rulerunner as a user.

About this task

You must use the same authentication method that is currently used by Datacap.

- [Creating or ensuring an account exists for Rulerunner](#)
You must create or ensure that a domain/Windows account exists for the Rulerunner Service.
- [Authenticate Rulerunner Service](#)
You must set up the Rulerunner Service to use one of the authentication systems that are supported by Datacap.
- [Authenticating Rulerunner using Datacap authentication](#)
You must prepare Rulerunner to use Datacap authentication.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Creating or ensuring an account exists for Rulerunner

You must create or ensure that a domain/Windows account exists for the Rulerunner Service.

About this task

Datacap does not require that a unique Windows account be set up for Rulerunner. Rulerunner can use any Windows account provided that account can be set up with the appropriate sharing and security permissions.

For example, all instances of Rulerunner can use the same account.

Parent topic: [Configuring Rulerunner authentication](#)

Authenticate Rulerunner Service

You must set up the Rulerunner Service to use one of the authentication systems that are supported by Datacap.

TMA authentication

You must do the following when Datacap is set up to use the TMA authentication method.

- Ensure that a Datacap user exists in your application that the Rulerunner Service can use. All instances of Rulerunner can use the same user. For details, see [Adding a Datacap user to your application](#).
- Ensure that a Datacap station exists in your application that the Rulerunner Service can use. All instances of Rulerunner can use the same station. For details, see [Adding a Datacap station to your application](#).
- In the Rulerunner Manager, set up each instance of Rulerunner to use Datacap Authentication and enter the Datacap user, password, and Datacap station information.

ADSI and LDAP authentication

You must do the following when Datacap is set up to use the ADSI or LDAP authentication method.

- Obtain the name of the domain in which Datacap and your authentication server are located. For details, see [Obtaining the name of the domain](#).
- Obtain the name of the Datacap Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) security group that is used to authenticate Rulerunner. For details, see [Obtaining the name of the AD/LDAP security group](#)
- Ensure that at least one domain/Windows account exists that the Rulerunner Service can use, and that the Windows account is part of the Datacap AD/LDAP security group. All instances of the Rulerunner Service can use the same Windows account. When you install and configure Rulerunner, be sure to use this Windows account for the Rulerunner Service.
- Add a Datacap group to your application with a name that corresponds to the AD/LDAP security group and short domain name. For details, see [Adding a Datacap group to your application for Rulerunner](#).
- Add a Datacap station for each Rulerunner server. The station name must be the same name as the Rulerunner server to your Datacap application. For details, see [Adding a Datacap station to your application for Rulerunner](#).
- In the Rulerunner Manager, set up each instance of Rulerunner to use Windows Authentication.

ADLDS and LLDAP authentication

You must do the following when Datacap is set up to use the ADLDS or LLDAP authentication method.

- Ensure that an account exists for the Rulerunner Service and that the account is set up in the ADLDS or LLDAP authentication system. All instances of the Rulerunner Service can use the same account.
- Ensure that a Datacap user exists that corresponds to the ADLDS or LLDAP account. All instances of Rulerunner can use the same user.
- Ensure that a Datacap station exists that the Rulerunner Service can use. All instances of Rulerunner can use the same station.
- In the Rulerunner Manager, set up each instance of Rulerunner to use Datacap Authentication and enter the Datacap user, password, and Datacap station information.

- [Obtaining the name of the domain](#)
When the Datacap Server Service is configured to use external AD (ADSI) or LDAP authentication, you need a short domain name of a domain. This domain is where the Datacap-specific Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) security group was created.
- [Obtaining the name of the AD/LDAP security group](#)
If you use ADSI or LDAP authentication, the domain/Windows account that is used by the Rulerunner Service must be part of a Datacap-specific Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) security group.
- [Logging in to Datacap Web Client](#)
You must log in to Datacap Web Client to authenticate Datacap groups, stations, and users to Rulerunner.
- [Adding a Datacap group to your application for Rulerunner](#)
You can add a Datacap group to your application for Rulerunner when you use either the AD (ADSI) or LDAP external authentication system.
- [Adding a Datacap station to your application for Rulerunner](#)
You can add a Datacap station to your application for Rulerunner when you use the AD (ADSI) or LLDAP external authentication system.
- [Adding a Datacap user to your application for Rulerunner](#)
You can add a Datacap user to your application for the Rulerunner Service when you are using ADLDS or LLDAP external authentication.

Parent topic: [Configuring Rulerunner authentication](#)

Obtaining the name of the domain

When the Datacap Server Service is configured to use external AD (ADSI) or LDAP authentication, you need a short domain name of a domain. This domain is where the Datacap-specific Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) security group was created.

About this task

You use the short domain name to set up the corresponding Datacap group in your Datacap application. You must know the exact spelling of the domain.

Parent topic: [Authenticate Rulerunner Service](#)

Obtaining the name of the AD/LDAP security group

If you use ADSI or LDAP authentication, the domain/Windows account that is used by the Rulerunner Service must be part of a Datacap-specific Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) security group.

About this task

Obtain the name of Datacap-specific Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) security group.

Parent topic: [Authenticate Rulerunner Service](#)

Logging in to Datacap Web Client

You must log in to Datacap Web Client to authenticate Datacap groups, stations, and users to Rulerunner.

Before you begin

Before logging in to Datacap Web Client, [start or ensure the Datacap Server Service is started](#).

Procedure

To log in to Datacap Web Client:

1. Open Internet Explorer, and enter the address of the Datacap web server (<http://WebServerName>). After a pause, the Datacap Login window opens.
Tip: You might have to turn off Internet Explorer popup blocker the first time you display the Login page.
2. On the Datacap Web Client Login page, select the appropriate application, enter your User ID, Password, and Station, then click Login.

Parent topic: [Authenticate Rulerunner Service](#)

Adding a Datacap group to your application for Rulerunner

You can add a Datacap group to your application for Rulerunner when you use either the AD (ADSI) or LDAP external authentication system.

About this task

The Datacap group name must correspond to the security group name set up in the authentication system.

Procedure

To add a Datacap group to your application:

1. On the Datacap Web Client Administrator tab, select the Groups tab.
2. Click New and replace the default text in the Name and Description fields. The Group ID for the Rulerunner Service requires the following structure.
 - o Your Datacap security group name, such as *TMUsers*.
 - o A dot
 - o The short name of the domain in which your Datacap security group is located, such as the *XYZ* domain.

In this example, *TMUsers.XYZ* would be the Datacap Group ID.

Each field can contain a maximum of 100 characters.

3. Since you are using this group to grant permissions to Rulerunner to process background tasks, under Permissions, select the appropriate background tasks in each job, then click Save group.

Parent topic: [Authenticate Rulerunner Service](#)

Adding a Datacap station to your application for Rulerunner

You can add a Datacap station to your application for Rulerunner when you use the AD (ADSI) or LLLDAP external authentication system.

About this task

Use the name of the Rulerunner server as the station name. You must enter the Rulerunner server name exactly as it was identified to the Domain Controller. For example, if the Rulerunner server name that is identified in the Domain Controller is in all capital letters, be sure to enter the station name in all capitals.

If you are setting up more than one Rulerunner server, repeat these instructions to add a station for each Rulerunner server.

Procedure

To add a Datacap station to your application:

1. On the Datacap Web Client Administrator tab, select Stations.
2. Click New and enter the name of the Rulerunner server name exactly as it was identified to the Domain Controller. Enter a Description.
3. Set the Maximum number of virtual stations to 9999, then click Save.
4. Since you are using the Rulerunner server as the station to run background jobs with Rulerunner, under Permissions, select the appropriate background tasks in each job, then click Save.

Parent topic: [Authenticate Rulerunner Service](#)

Adding a Datacap user to your application for Rulerunner

You can add a Datacap user to your application for the Rulerunner Service when you are using ADLDS or LLDAP external authentication.

About this task

The Datacap user name must correspond to the name of the account that you set up for Rulerunner in the ADLDS or LLDAP authentication system.

Procedure

To add a Datacap user to your application:

1. On the Datacap Web Client Administrator tab, select Users.
2. Click New and replace the default text in the Name field with the account name exactly as it was set up in the ADLDS or LLDAP authentication system. Enter the Description. Then, enter the password of the ADLDS or LLDAP account in both password fields and click Save user.
3. Because you are using this user to grant permissions to Rulerunner to process background tasks, under Permissions, select the appropriate background tasks in each job, then click Save user.

Parent topic: [Authenticate Rulerunner Service](#)

Authenticating Rulerunner using Datacap authentication

You must prepare Rulerunner to use Datacap authentication.

Before you configure Rulerunner Service to use Datacap authentication:

- Create or ensure that a domain/Windows account exists for the Rulerunner Service.
- Add a User ID and Station ID to your Datacap application for the Rulerunner Service.
- Configure the Rulerunner Service to use Datacap authentication.

- Grant the Rulerunner Service domain/Windows account the appropriate sharing and security permissions to the Datacap folders.

Parent topic: [Configuring Rulerunner authentication](#)

Installing and configuring the Rulerunner Service

You can install and configure the Rulerunner Service in an environment where Datacap is running successfully. Where the Workstation and Server are separate machines. And where you are adding one or more additional Rulerunner servers to host instances of the Rulerunner Service.

For the purpose of these instructions, the scenario includes computers in this configuration:

- A Windows Server 2008 hosts the Datacap Server, application files, batch files, and databases.
- One or more Windows 2008 Rulerunner servers each host an instance of the Rulerunner Service.
- One or more Workstations host either a browser or the Datacap Client software.
- At least one developer workstation hosts the complete Datacap Client software component.
- [Shutting down the Datacap software](#)
Before you start the installation and configuration process, you might want to identify which Datacap software is running so that you can restart it after you install Rulerunner.
- [Configuring Rulerunner security and permissions on the Datacap server](#)
You must set up sharing permissions and security for the Rulerunner Service domain/Windows account on various folders when the Datacap Server is running on Windows 2008.
- [Installing Rulerunner on the Rulerunner server](#)
You run the Datacap installation program wizard on the Rulerunner server to install the necessary Datacap software components.
- [Installing third party software components on Rulerunner server](#)
Each Rulerunner server that hosts Rulerunner requires that the appropriate Datacap and related third party software components be installed if you want Rulerunner to run tasks that use those components.
- [Importing encryption keys to Datacap computers](#)
In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.
- [Setting the location of the datacap.xml file](#)
You must define the location of the Datacap application settings by using the Datacap Application Manager. The datacap.xml file contains the application settings.
- [Granting permissions to the Rulerunner account on the Rulerunner server](#)
You must grant the Rulerunner domain/Windows account the appropriate permissions on its host Rulerunner server using DCOM Config.
- [Setting up security on the systemprofile\AppData folder for Rulerunner](#)
You can set up the appropriate security permissions for Rulerunner on the c:\Windows\SysWOW64\config\systemprofile\AppData folder on the Rulerunner server when the operating system is Windows 2008.
- [Granting Rulerunner the Log On as Service privilege](#)
You can ensure the domain/Windows account used by Rulerunner has been granted the Log On as a Service privilege on its Rulerunner server.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Shutting down the Datacap software

Before you start the installation and configuration process, you might want to identify which Datacap software is running so that you can restart it after you install Rulerunner.

About this task

Shut down all of the Datacap software that is running in your environment in the following sequence.

Procedure

To shut down the Datacap software:

1. Shut down all Datacap client software such as Datacap Desktop, Datacap Studio, Maintenance Manager, FastDoc, and all Datacap web clients.
2. Shut down Datacap Web Services and all other web services, such as Report Viewer, Fingerprint Service, and the Datacap Web Client Upload Service.
3. Shut down Datacap Server Service.
4. [Stop or ensure the Datacap Server Service is stopped.](#)

Parent topic: [Installing and configuring the Rulerunner Service](#)

Configuring Rulerunner security and permissions on the Datacap server

You must to set up sharing permissions and security for the Rulerunner Service domain/Windows account on various folders when the Datacap Server is running on Windows 2008.

- [Setting up sharing permissions for Rulerunner on the Datacap folder](#)
You must set up the appropriate sharing permissions for the Rulerunner Service account on the Datacap Server shared c:\Datacap folder.
- [Setting up security for Rulerunner on the Datacap folder](#)
You must set up the appropriate security for the Rulerunner Service account on the Datacap Server shared c:\Datacap folder.
- [Setting up security on the RRS folder](#)
You must set up the appropriate security permissions for the RRS folder on the Datacap Server when the Server is running on Windows 2008. The other accounts were granted security permissions during the installation and configuration of Datacap.
- [Setting up security on the Datacap\Application folder for Rulerunner](#)
You must set up the appropriate security permissions for Rulerunner on the c:\Datacap\Application folder on the Datacap Server when the Server is running on Windows 2008.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Setting up sharing permissions for Rulerunner on the Datacap folder

You must set up the appropriate sharing permissions for the Rulerunner Service account on the Datacap Server shared c:\Datacap folder.

About this task

These instructions apply when the Server's operating system is Windows 2008. Note that other accounts were already granted sharing permissions during the installation and configuration of Datacap.

Procedure

1. On the Server, start Windows Explorer, navigate to and right-click the c:\Datacap folder and select Properties.
2. Click the Sharing tab to display it. The folder should already be shared with the Share name of Datacap.
3. Then, click Advanced Sharing. When User Account Control (UAC) is on, the User Account Control window is displayed. Click Yes.
4. Click Permissions. Add or ensure that the domain/Windows User ID of the Rulerunner Service is set to allow Full Control.

Parent topic: [Configuring Rulerunner security and permissions on the Datacap server](#)

Setting up security for Rulerunner on the Datacap folder

You must set up the appropriate security for the Rulerunner Service account on the Datacap Server shared c:\Datacap folder.

About this task

These instructions apply when the operating system is Windows 2008. Note that other accounts were already granted security permissions during the installation and configuration of Datacap.

Procedure

1. On the Server, start Windows Explorer, navigate to and right-click the c:\Datacap folder and select Properties.
2. Click the Security tab to display it. Then, click Edit. When User Account Control (UAC) is on, the User Account Control window is displayed. Click Yes.
3. Add or ensure that the domain/Windows User IDs of Rulerunner Service account is set to allow Full Control.

Parent topic: [Configuring Rulerunner security and permissions on the Datacap server](#)

Setting up security on the RRS folder

You must set up the appropriate security permissions for the RRS folder on the Datacap Server when the Server is running on Windows 2008. The other accounts were granted security permissions during the installation and configuration of Datacap.

Procedure

To set up security on the RRS folder:

1. On the Server, start Windows Explorer, navigate to and right-click the c:\Datacap\RRS folder and select Properties.
2. Click the Security tab, then click Edit.
3. Add or ensure that the Domain/Windows User ID of the Rulerunner Service is set to allow Full Control.

Parent topic: [Configuring Rulerunner security and permissions on the Datacap server](#)

Setting up security on the Datacap\Application folder for Rulerunner

You must set up the appropriate security permissions for Rulerunner on the c:\Datacap\Application folder on the Datacap Server when the Server is running on Windows 2008.

About this task

If you are setting up Rulerunner to run tasks from more than one application, repeat these instructions for each C:\Datacap\Application folder, where *Application* is the application installation subdirectory. For example:

Datacap Medical Claims

C:\Datacap\Medical Claims\

Datacap Accounts Payable

C:\Datacap\APT\

Datacap TravelDocs

C:\Datacap\TravelDocs\

Note that other accounts were already granted security permissions during the installation and configuration of Datacap.

Procedure

1. On the Server, start Windows Explorer, navigate to and right-click the c:\Datacap\Application folder and select Properties.
2. Click the Security tab, then click Edit.
3. Add or ensure that the domain/Windows User ID of the Rulerunner Service is set to allow Full Control.

Parent topic: [Configuring Rulerunner security and permissions on the Datacap server](#)

Installing Rulerunner on the Rulerunner server

You run the Datacap installation program wizard on the Rulerunner server to install the necessary Datacap software components.

About this task

The components that you install include the Rulerunner software component, the Datacap Client software component, and the separately licensed Datacap connectors for which you have a license.

These instructions apply to servers that are running on Windows 2008.

If you have multiple Rulerunner servers, repeat this process for each server.

Procedure

To install Rulerunner on the Rulerunner server:

1. Make the installation package available on your network, or insert the Datacap CD in the Rulerunner server's CD/DVD drive. If the installation process does not start automatically, or if the package is on the network, open Windows Explorer, navigate to and double-click the Setup.exe. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.

2. Select the appropriate language, then click OK. The language that you select determines the language that is displayed by the installation program wizard during the installation process.
3. When more, redistributable software is required, the installation program wizard displays a list of the items to be installed. Click Install.
4. Click Next.
5. Accept the license agreement. Then, click Next.
6. Select the Custom option. Then, click Next.
7. Exclude all components from the installation process except the Datacap Client, Rulerunner, and the separately licensed connectors for which you have a license.
8. Click Next.
9. Click Install.
10. Click Finish.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Installing third party software components on Rulerunner server

Each Rulerunner server that hosts Rulerunner requires that the appropriate Datacap and related third party software components be installed if you want Rulerunner to run tasks that use those components.

When you installed Rulerunner, you were instructed to install Rulerunner as well as other Datacap software components.

If you are setting up Rulerunner to export to IBM® or third party repositories, ensure that you have installed whatever additional software is required by those repositories or third party software on each Rulerunner server.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Importing encryption keys to Datacap computers

In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.

Before you begin

You must generate the encryption keys in the keystore on a server on which the Datacap server software component is installed. You export the new keys to a key transport file.

Procedure

To import encryption keys to Datacap computers:

1. Find the dc_KTF.xml key transport file in the c:\Datacap\Taskmaster folder on the Datacap server where you generated and exported the encryption keys.
2. Copy the dc_KTF.xml key transport file to the appropriate folder on the computer where you installed the new component. The encryption keys are automatically applied to the keystore the next time you start the Datacap component.

Table 1. Encryption key folder locations by component

Component	Folder
Datacap Server	C:\Datacap\Taskmaster
Datacap applications	C:\Datacap\Taskmaster
Datacap Desktop	C:\Datacap\DcDesktop
Datacap Studio	C:\Datacap\DStudio
Datacap FastDoc	C:\Datacap\FastDoc
Datacap Report Viewer	C:\Datacap\RV2\bin
Datacap Web Client	C:\Datacap\tmweb.net\bin
Datacap Web Services	C:\Datacap\wTM\bin

Parent topic: [Installing and configuring the Rulerunner Service](#)

Setting the location of the datacap.xml file

You must define the location of the Datacap application settings by using the Datacap Application Manager. The datacap.xml file contains the application settings.

About this task

The datacap.xml file that contains the application settings is on the Datacap server.

Procedure

To set the location of the datacap.xml file:

1. In the Start menu, select IBM Datacap Services>Datacap Application Manager. If the User Account Control window opens, click Yes.
2. Click the Service tab to display it.
3. Change or ensure that the path reflects the correct location of the datacap.xml file. For example, \\Server\Datacap\datacap.xml
4. Close the Datacap Application Manager.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Granting permissions to the Rulerunner account on the Rulerunner server

You must grant the Rulerunner domain/Windows account the appropriate permissions on its host Rulerunner server using DCOM Config.

Procedure

To grant permissions to the Rulerunner account on the Rulerunner server:

1. From the Rulerunner server's Windows Start menu, select Administrative Tools > Component Services > Computers > My Computer > DCOM Config.
2. In the middle pane, locate, right-click the DCOProcessor application, then select Properties.

3. Click the Security tab to display it.
4. Under Launch and Activate Permissions, select Customize, then click Edit.
5. Add the Rulerunner domain/Windows account and set Local Launch and Local Activation to Allow.
6. Click OK.
7. Click OK.
8. In the middle pane, locate, right-click the RRProcessor application, then select Properties.
9. Click the Security tab to display it.
10. Under Launch and Activate Permissions, select Customize, then click Edit.
11. Add the Rulerunner domain/Windows account and set Local Launch and Local Activation to Allow.
12. Click OK.
13. Click OK.
14. Close the Component Services window.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Setting up security on the systemprofile\AppData folder for Rulerunner

You can set up the appropriate security permissions for Rulerunner on the c:\Windows\SysWOW64\config\systemprofile\AppData folder on the Rulerunner server when the operating system is Windows 2008.

About this task

If you are setting up multiple Rulerunner servers, repeat these instructions on each Rulerunner server.

Procedure

To set up security for Rulerunner:

1. On the Rulerunner server, start Windows Explorer, navigate to and right-click the c:\Windows\SysWOW64\config\systemprofile\AppData folder and select Properties.
2. Click the Security tab, then click Edit.
3. Add or ensure that the domain/Windows User ID of the Rulerunner Service is set to allow Modify.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Granting Rulerunner the Log On as Service privilege

You can ensure the domain/Windows account used by Rulerunner has been granted the Log On as a Service privilege on its Rulerunner server.

About this task

This privilege allows Datacap Rulerunner Service to run as a service.

If you have multiple Rulerunner servers, repeat this process on each server.

Procedure

To grant Rulerunner the Log On as Service privilege:

1. From the Rulerunner server's Windows Start menu, select Administrative Tools > Services. Then right-click Datacap Rulerunner Service and select Properties.
2. Click the Log On tab and select This account.
3. Click Browse and select the domain/Windows account for Datacap Rulerunner Service. Enter the password of the account twice, and click Apply.
4. Click OK to close the message box.
5. Click OK to close the Properties dialog.
6. Close the Services window.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Configuring Rulerunner to run your applications

You can configure one or more instances of Rulerunner to run background tasks from your applications or workflows.

About this task

Rulerunner can run a Datacap Studio application task that has the following characteristics:

- Rules-based
- Can be run without human interaction (and therefore does not need a user interface)

Tasks that are typically run by Rulerunner are recognition, image pre-processing, validation, and export tasks. Virtual scan tasks (VScan) can also be run if they are designed to pick up images automatically, rather than requiring a user to select images manually.

This section assumes that you have:

- Working versions of your Datacap applications, and that you have manually run all of the tasks successfully.
- Configured Rulerunner authentication by following the instructions in either [Authenticate Rulerunner Service](#) or [Authenticating Rulerunner using Datacap authentication](#).
- Installed Rulerunner following the instructions in [Installing and configuring the Rulerunner Service](#) on at least one Rulerunner server.

For the purposes of illustration, the configuration used in this scenario includes the following computers:

- The Server hosts the Datacap Server Service, the application files for one or more applications, batch files, and databases.
- One or more Rulerunner servers host Rulerunner
- One or more Workstations host either a browser or Datacap Client software for your applications.
- [Gathering information that you need to set up Rulerunner](#)
Before you begin setting up Rulerunner, you must gather the detailed information that is needed to set up Rulerunner.
- [Configuring the task profiles for Rulerunner to run](#)
Rulerunner runs the task profiles on the Datacap Application Manager. You must ensure the Datacap Application Manager lists the task profiles to be run by Rulerunner.
- [Configuring Rulerunner to run tasks](#)
You can set up one or more threads and configure Rulerunner to run tasks by using that thread. Repeat the instructions as many times as required to set up tasks from multiple applications.
- [Starting the Rulerunner Service](#)
After you set up or change the Rulerunner Service, go to the Rulerunner Manager and start the Rulerunner service.

- [Monitoring batches during Rulerunner processing](#)
Monitor your batches using the Datacap Web Job Monitor to watch batches change status as they are processed by Rulerunner.
- [Restart Datacap software](#)
After successfully installing, configuring, and by using Datacap Rulerunner to run more tasks, you can restart the Datacap software that was previously running on other computers.
- [Rulerunner thread configuration](#)
The number of Rulerunner threads to configure for an application depends on the type of tasks that your application performs.

Parent topic: [Installing and configuring the Rulerunner Service](#)

Gathering information that you need to set up Rulerunner

Before you begin setting up Rulerunner, you must gather the detailed information that is needed to set up Rulerunner.

Before you begin

You will need to [start or ensure the Datacap Server Service is started](#) in order to gather some of the information needed to set up Rulerunner.

About this task

For each of your applications or workflows, make a list that includes all of the following information. Be sure to copy the names of these items exactly because they are case-sensitive and the Rulerunner names must match.

Procedure

1. Identify whether Datacap is using Datacap authentication or Windows authentication. If Datacap is using Windows authentication, identify the type of authentication that being used, such as ADSI, LDAP, ADLDS, or LLLDAP.
2. Identify the externally visible names of your Rulerunner servers and your Datacap Server. These names are the server names that are maintained by your DNS.
3. Identify the number of processors available on each Rulerunner server.
4. For each of your Datacap applications, obtain the full UNC path and file names of the Administration Database and Engine Database.
5. For each of your Datacap applications, obtain the full path and folder name of the application or workflow folder on the server.
6. For each of your Datacap applications, obtain the workflow, job, and task name of each background task that you want Rulerunner to process.
 - a. To see these names, start your browser and log in to your application by using Datacap Web Client.
 - b. Select the Administrator tab, and make a note of the workflow name.
 - c. Expand the workflow, select the job, and make a note of the Job Name.
 - d. Expand the job, select the task, and make a note of the Task Name.
7. Identify the Task Profile Name of each background task that you want Rulerunner to process. Normally, the Task Profile Name is the same as the Task Name.
 - a. To see these names, start Datacap Studio, log in to your application and select the Test tab.
 - b. On the Workflow tab, expand the job, and make a note of the Task Profile Name.
8. Process documents that represent your typical production batches through your application. Identify approximately how long each potential Rulerunner task takes to process a typical batch. Identify any peaks or lulls that typically happen during specific time periods, such as daily, weekly, monthly,

quarterly. Identify whether your input documents must be placed into batches immediately, or if it input documents can be queued up without being turned into batches immediately. You must collect enough information to determine the batch creation volumes that your production system must process. This information helps you determine how many threads must be set up for each batch creation task and each background task so that your processing capability is maximized. For example, if you have an application that has one workflow that includes an email batch creation task and a background Profiler (PageID and recognition) task, run enough batches to gather the following type of information:

- a. Typical batches are processed by email batch creation in 5 seconds.
 - b. The same batches are processed by Profiler in 100 seconds.
9. You can experiment with increasing or decreasing the size of the batches that are produced by the email batch creation. If you are licensed to use multiple Rulerunner threads, you can calculate the number of threads that must be dedicated to the longest running tasks to process the pending batches in a timely manner. Depending on how quickly you want the newly created batches to be processed, you can have more threads for processing batches than for creating batches. Using the example in step 8, it takes 20 times longer for Profiler to process a batch than for your email batch creation task to create a batch. To maximize the number of batches that are processed, you might set up 20 Profiler threads for each email batch creation thread.
 10. You might also want to experiment with changing one or more of the available Datacap priority settings. Rulerunner priority settings are set at the task level using the Rulerunner Manager. Outside of Rulerunner, a batch priority can be assigned to batches produced by specific jobs and can be changed in the Datacap Web Client. On the Administrator tab, select the job and change the value in the Priority field. You can also enhance your applications to set or change priorities that are based on conditions or flags by using rules and the @PILOT(PRIORITY) Smart Parameter.
 11. Considering the number of threads that are allowed by your license agreement, and the number of processors available on each Rulerunner server, identify which tasks from the workflows or applications that you want to run on specific threads. For more information, see [Rulerunner thread configuration](#).

Parent topic: [Configuring Rulerunner to run your applications](#)

Configuring the task profiles for Rulerunner to run

Rulerunner runs the task profiles on the Datacap Application Manager. You must ensure the Datacap Application Manager lists the task profiles to be run by Rulerunner.

About this task

Before, you begin you must know which tasks and task profiles you want Rulerunner to process. You must have the exact spelling of each task name and task profile name. Gather the information that you need to set up Rulerunner before you continue with this procedure.

Procedure

To configure the task profiles for Rulerunner to run:

1. From the Windows Start menu on the developer workstation, select IBM Datacap Services>Datacap Application Manager. When User Account Control (UAC) is on, the User Account Control window opens, click Yes.
2. Select your application. Paths display in the fields on the Main tab.
3. Ensure that all of the paths are correct.
4. Click the Rulerunner tab. This tab displays only the task profiles that Rulerunner is to process.
5. To remove a task profile, click the red X next to the profile name. Then, click Yes to confirm you want to remove the task.
6. To add a task profile, click Add new Task.

7. Enter the name of the task in the first field and enter the name of the task profile in the second field. Ensure that the spelling and case are correct.
8. Close the Datacap Application Manager.
9. [Stop](#) and [restart](#) the Datacap Server Service.

Parent topic: [Configuring Rulerunner to run your applications](#)

Configuring Rulerunner to run tasks

You can set up one or more threads and configure Rulerunner to run tasks by using that thread. Repeat the instructions as many times as required to set up tasks from multiple applications.

About this task

Important: You can use this procedure to temporarily set up Datacap authentication and the Datacap `admin` user ID and password to make your configuration changes. Be sure to follow the final steps of this procedure to reset the user ID, password, and authentication method before you disconnect from the Datacap Application Service.

Procedure

To configure Rulerunner to run tasks:

1. From the Start menu, select IBM Datacap Services > Datacap Rulerunner Manager. If the User Account Control window opens, click Yes.
2. If the Status displayed is Stopped, continue to the next step. If the Status displayed is Running, click Stop.
3. Click the Rulerunner Login tab.
4. Select the Datacap Authentication option to enable the login credential fields.
5. Enter the User ID, `admin`, Password, `admin`, and Station ID, `1`. Click Connect.
6. Click the Workflow:Job:Task tab. The names of the applications from the `datacap.xml` file are displayed. The other pane does not contain threads the first time you use Rulerunner Manager.
7. Click the check box to select your application. The application tree expands with the server, the Administration database, and the Engine database selected.
8. Right-click in the other pane to display the menu items.
 - o Expand all - Expands the details of all existing threads.
 - o Collapse all - Collapses the details of all existing threads.
 - o Threads
 - Clear - Deletes all existing threads.
 - Add Thread - Adds a new, empty thread.
 - Add Threads - Adds multiple threads or copies a setting from an existing thread.
 - o Copy - Copies a selected thread.
 - o Paste - Adds a copied thread.
 - o Remove - Deletes the selected thread.
9. Right-click to select Threads and Add Thread. A new thread is added to the pane. For information about multiple-thread limitations, see [Rulerunner thread configuration](#).
10. To configure the individual thread to run one or more tasks, do one or more of the following actions.

Table 1. Thread configuration options

Option	Procedure
Configure a thread to run all job tasks in a workflow	Click the check box next to the workflow name, then click and drag the workflow name onto a thread in the other pane.

Option	Procedure
Configure a thread to run all tasks in a single job	Click the check box next to the job name, then click and drag the job name onto a thread in the other pane.
Configure a thread to run two or more tasks from a job	Click the check box next to each task, then click and drag the job onto a thread in the other pane.
Configure a thread to run a single task	Click the check box next to the task, then click and drag the task onto a thread in the other pane.
To configure a thread to run more than one job in the same application	Click the check box next to each job, then click and drag the workflow onto a thread in the other pane.
To configure a thread to run tasks from more than one application	Click the check box next to each task in each application, then click and drag each job onto a thread in the other pane.
Remove a task from a thread	In the other pane, right-click the task and select Remove to delete it.
Disable a thread	Click the check box next to the thread (the check mark disappears), or right-click the thread and select Remove to delete it entirely.
Remove a thread	Right-click the thread and select Remove.
Remove all threads	Select Threads and Clear.

11. Click Save or CTRL+S to save your changes. Click Yes to create and save the configuration file.
12. To set a batch processing task to a higher priority than a batch creation task, select the task. The task ID appears in the bottom pane.
13. Change the value in the priority field.
14. When the task is a batch creation task like VScan, increase the value for the skipsamebatch field.
15. Click the Settings tab to make settings changes.
16. Click Save or CTRL+S to save your changes.
17. Click the Logging tab. When you select the Number of Messages setting on the Quick Log tab, the same level of logging is automatically applied to the ATM Log, Rulerunner Log, and the RRS Log tabs.
18. Click the ATM Log tab and select the settings.
19. Click the Rulerunner Log tab and select the settings.
20. Click the RRS Log tab and select the settings.
21. Click Save to save your changes.
22. When you complete the changes, click the Rulerunner Login tab.
23. Click Disconnect.
24. You must complete the following steps to ensure that the authentication credentials for Rulerunner are set properly.
 - o Windows authentication - Select the Windows Authentication option.
 - o Datacap authentication - Select the Datacap Authentication option, enter the User ID of the Rulerunner domain/Windows account, the Password, and the name of the Rulerunner server as the Station ID.
25. Click Save.
26. Close the Rulerunner Manager window.

What to do next

From a workstation, start your Datacap client application using your admin user ID, password and station ID. Run your application or applications so that there are batches pending for the tasks that Rulerunner is configured to process.

Parent topic: [Configuring Rulerunner to run your applications](#)

Starting the Rulerunner Service

After you set up or change the Rulerunner Service, go to the Rulerunner Manager and start the Rulerunner service.

Procedure

Follow this procedure to start the Rulerunner Service.

1. Go to Start > IBM Datacap Services > Datacap Rulerunner Manager. If the User Account Control window opens, click Yes.
2. If the Status is Stopped, click Start. The Status changes to Running.
3. Close the Rulerunner Manager window.

Parent topic: [Configuring Rulerunner to run your applications](#)

Monitoring batches during Rulerunner processing

Monitor your batches using the Datacap Web Job Monitor to watch batches change status as they are processed by Rulerunner.

Parent topic: [Configuring Rulerunner to run your applications](#)

Related information:

[Troubleshooting Rulerunner](#)

Restart Datacap software

After successfully installing, configuring, and by using Datacap Rulerunner to run more tasks, you can restart the Datacap software that was previously running on other computers.

Restart Datacap software on all computers in the following sequence:

1. Datacap Server Service
2. Datacap Web Client and any Datacap Web Services, such as Report Viewer, wTM, and Fingerprint Service
3. Datacap Client software that includes Datacap Client, Datacap Web Client, Datacap Studio, and any of the other Datacap Clients

Parent topic: [Configuring Rulerunner to run your applications](#)

Rulerunner thread configuration

The number of Rulerunner threads to configure for an application depends on the type of tasks that your application performs.

Table 1. Thread considerations

Single threads for export tasks	When your application's tasks do one or more of the following activities, use one Rulerunner thread only: <ul style="list-style-type: none">• Export to a text file or a Microsoft Access database.
--	---

	<ul style="list-style-type: none"> • Export files at the same time to a downstream system (when it is not possible to predict the effect of multiple threads). <p>For both the Setup DCO and Fingerprint database, a locking mechanism exists to prevent two processes from performing updates at the same time. Nevertheless, for the following task activities, you might want to use a single thread to avoid delays that are caused by contention:</p> <ul style="list-style-type: none"> • Update field positions in the Setup DCO at run time (includes Intellocate). • Update a Fingerprint database.
<p>Single threads for other tasks</p>	<p>Depending on the circumstances, the following action libraries can be run in a single-threaded Rulerunner configuration only:</p> <ul style="list-style-type: none"> • DatacapBox actions • Email actions • Ewsml actions • Iml actions • Vscan actions <p>For example, each VScan task that looks in a particular folder for the input files must be set up to run on a single thread. Also, if an email or fax batch creation task is set up to look in a single mailbox, that task can run on one thread only. The following batch creation tasks can run concurrently in multiple threads:</p> <ul style="list-style-type: none"> • Vscan tasks from two different applications if the tasks look in two different input folders • Vscan tasks from a single application but in separate jobs if the tasks look in two different input folders • Email or fax tasks from a single application but in separate jobs if the tasks look in different mailboxes
<p>Multiple threads</p>	<p>Apart from those situations in which single threading is required, enable at least the same number of threads as there are processors in your Rulerunner server. In some cases, more threads might be possible without diminishing performance. The maximum number of threads is 150% of the number of processors.</p> <p>For example, when Rulerunner is installed on a quad core server, enable a minimum of four threads and a maximum of six threads.</p> <p>Restriction: The appropriate licensing is required to configure Rulerunner to run multiple threads.</p>

Parent topic: [Configuring Rulerunner to run your applications](#)

Installing and configuring the Datacap Fingerprint Service

The Datacap Fingerprint Service, with actions in the Autodoc actions library, supports the ability to cache and use the fingerprints from one or multiple Datacap applications simultaneously. The Fingerprint Service eliminates the fingerprint load time for all but the first batch that is processed by an application.

About this task

The first time that a match is requested by an application, the Fingerprint Service loads the fingerprints for that application. When more than one application is set up to use a single Fingerprint Service, the first time each

application requests fingerprint matching, the fingerprints for that application are loaded into memory. Subsequent requests for matching work with the fingerprints in the cache in memory. When a matching fingerprint is not found and the application creates a new fingerprint, that new fingerprint is loaded into the cache.

The Datacap Fingerprint Service is a web service that is based on Microsoft Internet Information Services (IIS). The service can service multiple client requests simultaneously. The Datacap Fingerprint Service cache is shared by all threads. Therefore, the single load of fingerprints into the cache for an application is all that is required even when multiple threads are running.

Any application that uses the FindFingerprint action (from the Autodoc.RRX action library) to perform fingerprint matching can use the Fingerprint Service. To use the Fingerprint Service, add the SetFingerprintWebServiceURL action to the application at the batch level before you call the SetFingerprintDir action.

The Datacap Fingerprint Service is installed as part of the Rulerunner component and requires additional licensing.

This section provides the information that you need to install and configure the Datacap Fingerprint Service in a client/server environment.

- [Fingerprint Service configuration assumptions](#)
To install, configure, and test the configuration of the Fingerprint Service, you must ensure that certain requirements are met.
- [Creating or ensuring a Fingerprint Service account exists](#)
Create or ensure a domain/Windows account exists for the Fingerprint Service.
- [Setting up sharing permissions for the Fingerprint Service on the Datacap folder](#)
You must set up the appropriate sharing permissions for the Fingerprint Service account on the Datacap Server's shared c:\Datacap folder.
- [Setting up security on the Datacap\application\fingerprint folder for the Fingerprint Service](#)
Set up the appropriate security permissions for the Fingerprint Service account. The security permissions must be set up on the Datacap Server c:\Datacap\application\fingerprint folder, such as c:\Datacap\TravelDocs\batches.
- [Setting up security on the Datacap\application\batches folder for the Fingerprint Service](#)
Set up the appropriate security permissions for the Fingerprint Service account. The security permissions must be set up on the Datacap Server c:\Datacap\application\batches folder, such as c:\Datacap\TravelDocs\batches.
- [Installing the Fingerprint Service on the Fingerprint Service server](#)
Run the Datacap installation wizard on the Fingerprint Service server to install software components. Install the Rulerunner software component that contains the Fingerprint Service software.
- [Setting up security on the Datacap\FingerprintService folder for the Fingerprint Service](#)
This procedure provides instructions on how to set up the appropriate security permissions for the Fingerprint Service account on the c:\Datacap\FingerprintService folder on the Fingerprint Service server.
- [Adding the Fingerprint Service account to the IIS_IUSRS group](#)
You must add the domain/Windows account used by the Fingerprint Service to the IIS_IUSRS group on its host Fingerprint Service server.
- [Adding an Application Pool for the Fingerprint Service](#)
This procedure provides instructions on how to add a Microsoft Internet Information Services (IIS) Application Pool on the Fingerprint Service server for use by Fingerprint Service.
- [Setting up the Fingerprint Service on the Fingerprint Service server](#)
Set up the Fingerprint Service on a web server with Microsoft Internet Information Services (IIS) 7.5.
- [Validating the Fingerprint Service installation](#)
This procedure provides instructions on how to confirm that the Datacap Fingerprint Service is configured properly.

- [Verifying that the Fingerprint Service can load fingerprints from your application](#)
You can verify that the Datacap Fingerprint Service can load fingerprints from your application by using the Try Fingerprint Service tool and confirming that it loaded the correct fingerprints during the test.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Fingerprint Service configuration assumptions

To install, configure, and test the configuration of the Fingerprint Service, you must ensure that certain requirements are met.

Before you install the Fingerprint Service, ensure that the following conditions are met.

- All of the applications that you want the Fingerprint Service to support must be installed, configured, and run successfully without the Fingerprint Service.
- The environment in which these applications are run is an environment where the workstation and server are separate machines.
- You add a Fingerprint Service server to host the Fingerprint Service.

The scenario includes computers in the following configuration:

- A Windows server hosts the Datacap Server, application files, batch files, and databases.
- A Windows Fingerprint Service server hosts Microsoft Internet Information Services (IIS) and the Fingerprint Service.
- At least one developer workstation hosts the complete Datacap client software component, including Datacap Studio.

Before you begin installing the Fingerprint Service, ensure the following conditions.

- You have a working Datacap client/server environment.
- Datacap runs correctly in a client/server environment where the application files are on the Datacap Server and the application runs correctly from a workstation. This step makes troubleshooting easier, and narrows down the sources of potential problems.
- If you want the Fingerprint Service to cache fingerprints from two or more Datacap applications, ensure that those applications are working. In addition, ensure that you manually process batches from start to finish.
- When you install on client and server computers, ensure that you have administrator access to all computers on which Datacap software is installed. In addition, ensure that you have administrator access to the Windows server on which you intend to install the Fingerprint Service.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Creating or ensuring a Fingerprint Service account exists

Create or ensure a domain/Windows account exists for the Fingerprint Service.

About this task

Datacap does not require you to set up a unique Windows account for the Fingerprint Service. The Fingerprint Service can use any Windows account that you can set up with the appropriate sharing and security permissions. When the Fingerprint Service is installed on the same web server as Datacap Web Client or Report Viewer, the Fingerprint Service can use the same Windows account or a different one.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Setting up sharing permissions for the Fingerprint Service on the Datacap folder

You must set up the appropriate sharing permissions for the Fingerprint Service account on the Datacap Server's shared c:\Datacap folder.

About this task

These instructions apply when the Server's operating system is Windows 2008. Other accounts were already granted sharing permissions during the installation and configuration of Datacap.

Procedure

1. On the Server, start Windows Explorer, navigate to and right-click the c:\Datacap folder and select Properties.
2. Click the Sharing tab to display it. The folder should already be shared with the Share name of Datacap.
3. Then, click Advanced Sharing. When User Account Control (UAC) is on, the User Account Control window is displayed. Click Yes.
4. Click Permissions. Add or ensure that the domain/Windows user ID of the Datacap Fingerprint Service is set to allow Full Control.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Setting up security on the Datacap\application\fingerprint folder for the Fingerprint Service

Set up the appropriate security permissions for the Fingerprint Service account. The security permissions must be set up on the Datacap Server c:\Datacap\application\fingerprint folder, such as c:\Datacap\TravelDocs\batches.

About this task

Other accounts were already granted security permissions during the installation and configuration of Datacap.

Procedure

1. On the server, start Windows Explorer, go to the c:\Datacap\application\fingerprint folder, where *application* is the name of your application. Right-click, and select Properties.
2. Click the Security tab, then click Edit.
3. Add or ensure that the domain/Windows user ID of the Fingerprint Service is set to allow Read & Execute.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Setting up security on the Datacap\application\batches folder for the Fingerprint Service

Set up the appropriate security permissions for the Fingerprint Service account. The security permissions must be set up on the Datacap Server c:\Datacap\application\batches folder, such as

c:\Datacap\TravelDocs\batches.

About this task

Other accounts were already granted security permissions during the installation and configuration of Datacap.

Procedure

1. On the server, start Windows Explorer, go to the c:\Datacap*application*\batches folder, where *application* is the name of your application. Right-click, and select Properties.
2. Click the Security tab, then click Edit.
3. Ensure that the domain/Windows user ID of the Fingerprint Service is set to allow Read & Execute.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Installing the Fingerprint Service on the Fingerprint Service server

Run the Datacap installation wizard on the Fingerprint Service server to install software components. Install the Rulerunner software component that contains the Fingerprint Service software.

Procedure

1. Make the installation package available on your network, or insert the Datacap CD in the Fingerprint Service server CD/DVD drive. If the installation process does not start automatically, or if the package is on the network, open Windows Explorer, go to Setup.exe. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.
2. Select the appropriate language, then click OK. The language that you select determines the language that is displayed by the installation wizard during the installation process.
3. When additional software is required, the installation wizard displays a list of the items to be installed. Click Install.
4. Click Next.
5. Accept the license agreement. Then, click Next.
6. Select the Custom option. Then, click Next.
7. Exclude all components from the installation process except the Rulerunner component.
8. Click Next.
9. Click Install.
10. Click Finish.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Setting up security on the Datacap\FingerprintService folder for the Fingerprint Service

This procedure provides instructions on how to set up the appropriate security permissions for the Fingerprint Service account on the c:\Datacap\FingerprintService folder on the Fingerprint Service server.

Procedure

1. On the Fingerprint Service server, start Windows Explorer, go to the c:\Datacap\FingerprintService folder, and right-click to select Properties.
2. Click the Security tab, then click Edit.
3. Add the NETWORK SERVICE and local IUSR and set both to allow Read & Execute.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Adding the Fingerprint Service account to the IIS_IUSRS group

You must add the domain/Windows account used by the Fingerprint Service to the IIS_IUSRS group on its host Fingerprint Service server.

Procedure

1. From the Fingerprint Service server Windows Start menu, select Administrative Tools > Computer Management, expand Local Users and Groups, and then select Groups.
2. In the middle pane, right-click the IIS_IUSRS group, and then select Properties.
3. Add the Fingerprint Service domain/Windows account, then click OK to close the Properties window.
4. Close the Computer Management window.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Adding an Application Pool for the Fingerprint Service

This procedure provides instructions on how to add a Microsoft Internet Information Services (IIS) Application Pool on the Fingerprint Service server for use by Fingerprint Service.

Procedure

1. From the Windows Start menu, select Administrative Tools > Internet Information Services (IIS) Manager.
2. In the Connections pane, expand the computer, right-click Application Pools and select Add Application Pool.
3. Set the Name to `fp-service`.
4. Set the .NET Framework version to `.NET Framework v4.0.30319`.
5. Set the Managed pipeline mode to `Integrated`.
6. Select the Start application pool immediately option, then click OK.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Setting up the Fingerprint Service on the Fingerprint Service server

Set up the Fingerprint Service on a web server with Microsoft Internet Information Services (IIS) 7.5.

Procedure

To set up the Fingerprint Service on the Fingerprint Service server:

1. From the Windows Start menu, select Administrative Tools > Internet Information Services (IIS) Manager.

2. In the Connections pane, expand the computer and expand Sites. Right-click the Default Web Site, and then select Add Application.
3. Set the Alias to `fp-service`.
4. Click Select and select the `fp-service` Application Pool that you added, then click OK.
5. Set the Physical path by entering or browsing to the installation folder for the Fingerprint Service. The default location is `C:\Datacap\FingerprintService`.
6. Click OK to close the Add Application dialog.
7. Click Test Settings.
 - o If the test of the connection fails, click Close. Ensure that the permissions for the domain/Windows account are set up correctly, and that the application pool is set up correctly. Then, begin this procedure again.
 - o If the test of the connection is successful, click Close, then click OK and continue with the next step.
8. In the Connections pane, select Application Pools.
9. In the Application Pools pane, select the `fp-service` application pool.
10. In the Actions pane, click Advanced Settings.
11. Ensure that Enable 32-Bit Applications is set to True.
12. In the Process Model section, click Browse next to Identity.
13. In the Application Pool Identity window, select Custom account and click Set.
14. In the Set Credentials window, enter the Fingerprint Service domain/Windows account information (the same account that you added to the Fingerprint Service server's IIS_IUSRS group) in the format: `accountname@domainname`, enter the account password twice, then click OK.
15. In the Process Model section, set Idle Time-out to zero.
16. Click OK.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Validating the Fingerprint Service installation

This procedure provides instructions on how to confirm that the Datacap Fingerprint Service is configured properly.

Procedure

On the Fingerprint Server, from the Windows Start menu, start Internet Explorer and connect to the Fingerprint Service test page by entering: `http://127.0.0.1/fp-service/Service.asmx?WSDL`

Results

If the Service CCO Fingerprints DB Service page is displayed, the Fingerprint Service is configured properly.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Verifying that the Fingerprint Service can load fingerprints from your application

You can verify that the Datacap Fingerprint Service can load fingerprints from your application by using the Try Fingerprint Service tool and confirming that it loaded the correct fingerprints during the test.

Before you begin

Before you begin the verification, gather the following information:

- The full URL of the Fingerprint Service, for example:

`http://127.0.0.1/fpservice/Service.asmx?WSDL`

- The name of your application, for example: TravelDocs
- The full UNC path to fingerprint directory of your application, for example: \\ServerName\Datacap\TravelDocs\fingerprint
- The number of fingerprints currently in the fingerprint directory
- The names of some or all of the fingerprint files (.cco files) currently in the fingerprint directory

Procedure

To verify that the Fingerprint Service can load fingerprints from your application:

1. On the Fingerprint Service server, from the Windows Start menu, select IBM Datacap Developer Tools > Datacap Fingerprint Service Test Tool .
2. Enter the name of your application in the Application Name field.
3. Ensure that the full URL in the Fingerprints URL field is correct. An example of a full URL is `http://127.0.0.1/fpservice/Service.asmx?WSDL`.
4. Enter the full UNC path to fingerprint directory of your application in the Fingerprint Directory field, then click Upload All Fingerprints From Directory.
5. Confirm that the number in the Fingerprints Loaded field is correct, and that the names of the fingerprint files that are displayed in the pane are correct.
6. Confirm that the Fingerprint Service can match a fingerprint (.cco file) to one of the loaded fingerprints. In the Find Fingerprint field, browse to or enter the full path to one of the fingerprints that is displayed, and then click Find Fingerprint. The results of the fingerprint search and matching process are displayed. The match results are displayed in the format: '1.00;1;0;0' where:
 - The first number (1.00;) is from 0.0 to 1.00 where 0.0 means no match was found, and 1.00 means that an exact match was found.
 - The second number (1;) is the index of the matching fingerprint in the Fingerprint database.
 - The third and fourth numbers (0;0) are the X and Y offset of the test image relative to the stored fingerprint.
7. Close the Try Fingerprint Service window.

Parent topic: [Installing and configuring the Datacap Fingerprint Service](#)

Installing and configuring the Datacap Web Client upload service

After you scan files and create batches with Datacap, you can upload the files by using the Datacap Web Client upload service.

About this task

You can upload the files immediately or schedule an upload for a specific time. You can configure the service to upload the files at a time when network usage is less busy.

You can upload batches with the Datacap Web Client upload service in the following ways:

- Upload batches that are scanned by using the Datacap Web Client.
- Upload batches that are configured to use strict upload. With this method, batches initially are created on the client with Datacap Desktop, Rulerunner, or FastDoc. The Datacap Web Client upload service processes strict upload batches by creating batch folders in a shared location and then transfers images

and the page file. During the scan, the images are stored locally by Datacap Desktop, Rulerunner, or FastDoc, then the Datacap Web Client upload service uploads them.

- Upload batches with FastDoc by setting the StrictUpload parameter to False and entering an FDPATHMASK. The Datacap Web Client upload service runs a batch-creation task, which creates a batch and then uploads images and XML files. The Datacap Web Client upload service sets the batch level variable FDBatchID to track the original batch number. It then writes a line for each uploaded batch in the parent batch in the dir \dc2run.log file. After the batch is successfully processed, the FastDoc page file is removed.

Procedure

Follow this procedure to set up the Datacap Web Client upload service to run Datacap Web Client uploads in the background on remote scan stations.

1. Ensure that you are using the correct Datacap software version on all servers and workstations.
2. Install the Datacap files.
3. Edit the configuration file to match your environment and application. Configure the intervals at which you want the Datacap Web Client upload service to run background uploads.
4. Copy the files to each remote scan station. Edit individual configuration files to reflect the correct scan operator user ID, password, and station ID. Install the service on each scan station.
5. Configure the service to start automatically and start the service on each remote scan station.
6. Begin or continue scanning.

The Datacap Web Client upload service uses the Local Account to access scanned batches for that scan station. The upload service accesses the scanned batches in the Datacap Web Client Scan Into folder. The service also uses that account to upload the scanned batches and delete the batches from that folder after a successful upload.

If the Scan Into folder is not on the scan station, configure or ensure there is a domain account that Datacap Web Client Service can use. Set up the appropriate sharing, sharing permissions, and folder security, and grant the Datacap Web Client Service domain account Full Control on the Scan Into folder.

- [Editing the configuration file](#)
The Datacap Web Client upload service configuration file is installed in the Datacap folder after you configure the system.
- [Installing the Datacap Web Client upload service](#)
You can install the Datacap Web Client upload service from the Datacap server, or by copying the setup.exe file from another computer.
- [Configuring applications for the Datacap Web Client upload service](#)
After you install the Datacap Web Client upload service, you must specify which application to use when you upload batches.
- [Uploading batches scanned by using Datacap Web Client](#)
To upload batches that are scanned by using Datacap Web Client, you must change the dc2Run.exe.config file.
- [Uploading batches stored locally by connecting to the Datacap server](#)
When you use Datacap Desktop, Rulerunner, or FastDoc, you can create and store batches locally when you scan while your system is connected to the Datacap server.
- [Uploading completed offline batches by using FastDoc](#)
When you upload batches by using Datacap Web Client or by using StrictUpload=True, you use a scan task and an upload task.
- [Using Datacap Web Client upload service with Datacap Desktop](#)
This section provides information about how to create a working configuration for using Datacap Desktop, storing the scanned image files locally, and then uploading them automatically to the Datacap Server using the Datacap Web Client Upload Service.

- [Starting the Datacap Web Client upload service](#)
You configure the Datacap Web Client upload service differently depending on where the Scan Into is located.
- [Viewing the Event Log](#)
If you do not get the results you expect when you use the Datacap Web Client upload service, view the Application level log in the Event Viewer for the scan station. The log is where any issues or successfully completed tasks are logged.
- [Disabling the Datacap Web Client upload service](#)
You can disable the Datacap Web Client upload service if you are not using the service.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Editing the configuration file

The Datacap Web Client upload service configuration file is installed in the Datacap folder after you configure the system.

About this task

You configure the system by running the \support\webclientservice\setup.exe script. The configuration file is an XML file that contains default name-value pairs that must be changed to match your environment and your requirements.

Ensure that you have a backup copy of the original dc2run.exe.config file. Some of the required changes are global and affect all copies of the configuration file. Other changes are unique to each scan station. Make the global changes to a master copy of the configuration file before you copy it to the individual scan stations and make the station-specific changes.

- Before copying the configuration file to each remote scan station, change the settings required for your environment and application. Set up the intervals at which you want the Datacap Web Client Service to run background uploads.
- After copying the file to each individual remote scan station, edit the correct scan operator, user ID, password, and station ID. The password is saved by using the Datacap Application Manager.

Procedure

To edit the configuration file:

1. Open the dc2Run.exe.config file in Notepad or an XML editor.
 2. Before copying the configuration file to each remote scan station, change the settings required for your environment and application. Set up the intervals at which you want the Datacap Web Client Service to run background uploads.
 3. After copying the file to each individual remote scan station, edit the correct scan operator, user ID, password, and station ID. The password is saved by using the Datacap Application Manager.
- An example of the dc2run.exe configuration file. Some line breaks are used to fit the printed page.

```
<?xml version="1.0"?>
<configuration>
  <configSections>
    <sectionGroup name="applicationSettings"
type="System.Configuration.ApplicationSettingsGroup, System,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089">
      <section name="dc2Run.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089">
```

```

requirePermission="false"/>
  </sectionGroup>
</configSections>
<applicationSettings>
  <dc2Run.Properties.Settings>
    <setting name="dc2Run_localhost_utility" serializeAs="String">
      <value>http://127.0.0.1/Task/utility.asmx</value>
    </setting>
    <setting name="Application" serializeAs="String">
      <value>APT</value>
    </setting>
    <setting name="User" serializeAs="String">
      <value>admin</value>
    </setting>
    <setting name="Delay" serializeAs="String">
      <value>3000</value>
    </setting>
    <setting name="RetryDelay" serializeAs="String">
      <value>15000</value>
    </setting>
    <setting name="Station" serializeAs="String">
      <value>1</value>
    </setting>
    <setting name="ShortcutIndex" serializeAs="String">
      <value>13</value>
    </setting>
    <setting name="WorkTimes" serializeAs="String">
      <value>15:00-15:00</value>
    </setting>
    <setting name="StrictUpload" serializeAs="String">
      <value>False</value>
    </setting>
    <setting name="FDPathMask" serializeAs="String">
      <value>C:\Datacap\FastDoc\batches\APT\*\finished.xml</value>
    </setting>
  </dc2Run.Properties.Settings>
</applicationSettings>
<startup><supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
</startup></configuration>

```

Parent topic: [Installing and configuring the Datacap Web Client upload service](#)

Installing the Datacap Web Client upload service

You can install the Datacap Web Client upload service from the Datacap server, or by copying the setup.exe file from another computer.

Procedure

To install the Datacap Web Client upload service:

1. On the Datacap server, navigate to the following folders: Datacap > Support > Web Client Service.
2. Run the setup.exe file.
3. At the User Account Control window, click Yes to start the installation.
4. Click Next to continue the installation program.
5. Create a Web Client upload service destination folder and then click Next.
6. Click Install.
7. Click Finish when the installation is complete.

Note:

When WTM is hosted as a Windows service, an `IOException` is thrown when either the temporary file count exceeds 65535 within the user's temporary folder, or if no unique temporary file name is available. To avoid this exception, any unnecessary temporary files created by the "original file name" response variable are deleted from memory.

Parent topic: [Installing and configuring the Datacap Web Client upload service](#)

Configuring applications for the Datacap Web Client upload service

After you install the Datacap Web Client upload service, you must specify which application to use when you upload batches.

About this task

You must also specify the user password for that application and configure these settings on every scanning client where you installed the Datacap Web Client upload service.

Tip: You must configure your applications regardless of the method that you use to upload batches:

- Uploading batches by using Datacap Web Client
- Uploading batches by using a strict upload
- Uploading batches by using FastDoc

Procedure

To configure applications to use the Web Client upload service

1. From the Datacap folder, open the `dc2Run.exe.config` file in a text editor.
2. For the setting name parameter, enter the name of the application that you will use to upload the batch files. For example, if you are uploading batch files to the APT application, enter `APT`.

```
<setting name="Application" serializeAs="String">  
  <value>APT</value>  
</setting>
```

3. Set the user password that is used to authenticate the Datacap application.
 - a. In Datacap Application Manager, open the application that you are uploading the batches to, such as APT.
 - b. From the Custom Values tab, enter the Value name `dc2run.User` in the Advanced Values section. The value for User must be the Datacap user and must be the same user in the `dc2Run.exe.config` file. The username is case sensitive. For example, if the user is `admin`, the name must be `dc2run.admin`. In the case of a domain user, for example `DEVAD002\ScanUpload`, the name must be `dc2run.DEVAD002\ScanUpload`.
 - c. In Value field, enter the user password. For example, enter `admin`. The value name 1: in the Advanced values section must match the user that is specified in the `dc2Run.exe.config` file.
 - d. Click Add Now.
4. From the `dc2Run.exe.config` file, configure the following settings:

`dc2Run_localhost_utility`

Change the value of the URL value to reflect the IP address or UNC name of the Datacap Web Client server and path to `utility.asmx`.

```
<setting name="dc2Run_localhost_utility" serializeAs="String">  
<value>http://localhost/tmweb.net/Task/utility.asmx</value>
```

```
</setting>
```

http

Change this setting to `https` if you are using SSL. This change is made in the master copy of the configuration file.

```
<setting name="dc2Run_localhost_utility" serializeAs="String">  
<value>http://localhost/tmweb.net/Task/utility.asmx</value>  
</setting>
```

Application

The name of the application to be processed by the Datacap Web Client. This change is made in the master copy of the configuration file.

```
<setting name="Application" serializeAs="String">  
<value>TravelDocs</value>  
</setting>
```

User

The Datacap User for this application. This change is made in the scan station-specific copy of the configuration file.

```
<setting name="User" serializeAs="String">  
<value>admin</value>  
</setting>
```

Tip: You can also specify a domain user. For example,

```
<value>DEVAD002\ScanUpload</value>.
```

Station

The Datacap Station ID of the User. This change is made in the scan station-specific copy of the configuration file.

```
<setting name="Station" serializeAs="String">  
<value>remote</value>  
</setting>
```

ShortcutIndex

The `btn_ButtonNumber` value for the upload task. This change is made in the master copy of the configuration file.

```
<setting name="ShortcutIndex" serializeAs="String">  
<value>3</value>  
</setting>
```

To determine the appropriate `ShortcutIndex` value to use:

- a. On the Datacap Server, open Windows Explorer and navigate to the Admin database for the application, which is typically in the application `\process` folder.
- b. Use the appropriate database Client software to open the Admin database.
- c. Open the Buttons table, find the upload task in the `btn_ButtonName` column, and find the number for the upload task in the `btn_ButtonNumber` column.

Delay

The amount of time for the service to wait after finishing a successful upload and beginning the next upload. The value is in milliseconds. This change is made in the master copy of the configuration file.

```
<setting name="Delay" serializeAs="String">  
<value>3000</value>  
</setting>
```

RetryDelay

The amount of time for the service to wait between attempts to complete an upload. The value is in milliseconds. This change is made in the master copy of the configuration file.

```
<setting name="RetryDelay" serializeAs="String">
<value>15000</value>
</setting>
```

WorkTimes

Defines the StartTime and StopTime pairs in which the service is to run. Each Start and Stop pair is defined by using a 24 hour clock or 12 hour clock. You can use a mix of both types of clocks. Define as many pairs as you need. Within a pair, be sure the StartTime is earlier than the StopTime, for example:

- `<value>15:02-15:04</value>`: An interval defined by using the 24-hour clock
- `<value>3:05 PM-3:07 PM</value>`: An interval defined by using the 12-hour clock
- `<value>15:58-16:00;4:06 PM-4:08 PM</value>`: An interval defined by using the 24-hour clock followed by an interval defined by using the 12-hour clock. To run the service around the clock, use: `<value>0:00-23:59</value>`.

```
<setting name="WorkTimes" serializeAs="String">
<value>14:25-14:27;2:29 PM-2:29 PM</value>
</setting>
```

Important: The default StartTime and StopTime value is 15:00-15:00. Be sure that you change the default to an appropriate value for your environment.

StrictUpload

Defines how the batch files are uploaded. Change the value based on the uploading method:

- Set the StrictUpload parameter to True when you upload batches that are created locally by using Datacap Desktop, Rulerunner, or FastDoc on to distributed systems with a slow or irregular network connection.
- Set the StrictUpload parameter to False when you upload offline batches by using FastDoc.

```
<setting name="StrictUpload" serializeAs="String">
<value>True</value>
</setting>
```

FDPathMask

This value is based on how you upload batches. Clear the FDPathMask setting when you upload batches with Datacap Web Client or if you are uploading batches that are created locally by using Datacap Desktop, Rulerunner, or FastDoc. You must set the FDPathMask setting when you upload offline batches by using FastDoc.

```
<setting name="FDPathMask" serializeAs="String">
<value>c:\\datacap\FastDoc\batches\*\*\finished.xml</value>
</setting>
```

5. Save and close the dc2Run.exe.config file.

Parent topic: [Installing and configuring the Datacap Web Client upload service](#)

Uploading batches scanned by using Datacap Web Client

To upload batches that are scanned by using Datacap Web Client, you must change the dc2Run.exe.config file.

Procedure

To configure the Web Client upload service to upload batches that are scanned with Datacap Web Client:

Open the dc2Run.exe.config file in a text or XML editor and change the following settings:

ShortcutIndex

Change the value of ShortcutIndex setting to reflect the btn_ButtonNumber for the upload task. The change is made in the master copy of the configuration file.

```
<setting name="ShortcutIndex" serializeAs="String">  
<value>3</value>  
</setting>
```

To determine the appropriate ShortcutIndex value to use:

- a. On the Datacap Server, open Windows Explorer and navigate to the Admin database for the application, which is typically in the application \process folder.
- b. Use the appropriate database Client software to open the Admin database.
- c. Open the Buttons table, find the upload task in the btn_ButtonName column, and find the number for the upload task in the btn_ButtonNumber column.

StrictUpload

Set the StrictUpload setting to False.

```
<setting name="StrictUpload" serializeAs="String">  
<value>False</value>  
</setting>
```

FDPATHMask

Remove the value for the FDPATHMask setting.

```
<setting name="FDPATHMask" serializeAs="String">  
<value></value>  
</setting>
```

Parent topic: [Installing and configuring the Datacap Web Client upload service](#)

Uploading batches stored locally by connecting to the Datacap server

When you use Datacap Desktop, Rulerunner, or FastDoc, you can create and store batches locally when you scan while your system is connected to the Datacap server.

About this task

You can use the Web Client upload service to upload those batches to the Datacap server. You must copy the Application file locally to each scan client where you want to use the dc2run script and set the StrictUpload setting to True. For example, the local version of the APT.app application uses the C:\Datacap\APT\batches directory. The main versions used by Datacap Web Client and the verified clients use the shared location. You must duplicate the Application file locally on the scan client so that you can change the batch folder location.

Procedure

To connect with Datacap directly, and upload the images stored locally:

1. On the scanning workstation, open the dc2run.exe.config file.
2. Set the following setting: StrictUpload=True.
3. Remove the FDPATHMask value.

4. Copy your application file, such as \\myServer\myShare\datacap.xml, to a different shared directory to be used by all scan stations or copy it to each scan station. Add the application to the datacap.xml file. The application entry in the datacap.xml file, such as TravelDocs, must point to the correct location for TravelDocs.app. If the application file is in a subfolder with the same name, you can use a relative reference. Otherwise, you can enter the full path, for example:

```
<app name="APT" ref="APT"></app>  
<app name="TravelDocs" ref="c:\datacap\TravelDocs"></app>
```

5. Copy your application file, such as \\myServer\myShare\TravelDocs\TravelDocs.app, to a different shared directory to be used by all scan stations or copy it to each scan station to change the batch folder location.
6. Open the Datacap Application Manager, and click your application.
7. From the Service tab, change the Path to the application management file field to a path to the shared directory for the scan station clients or to a local path. The Path to the application management file setting currently points to the datacap.xml file, which populates the list of application names on the Datacap Application Manager window. Changing the path eliminates changing the settings on each client. The settings are all saved in one place.
8. From the Main tab, change the Batch folder to point to the local directory where files are to be stored prior to uploading.
9. Enable Create Batch Directory in Datacap Web Client.
 - a. Start Datacap Web Client and log in to the application that contains the Upload task.
 - b. Select Administrator > Workflow and the job that contains the Upload task.
 - c. Select Upload > Setup > Create Batch Directory and click Save.

Results

When you run the upload task, the upload task creates a batch folder in the shared location and moves files to it. So the Application file must be saved locally to the Datacap Desktop, Rulerunner, or FastDoc station.

Parent topic: [Installing and configuring the Datacap Web Client upload service](#)

Uploading completed offline batches by using FastDoc

When you upload batches by using Datacap Web Client or by using `StrictUpload=True`, you use a scan task and an upload task.

About this task

One is a scan task for batch creation, and one is an upload task to upload the batches. However, this scenario, uploading completed batches offline by using FastDoc, uses only one task, a scanning task. Because there is no separate upload task, you must go to the Datacap workflow in Datacap Web Client and set the batch creation task to use the `uplBFcl.aspx` file.

Procedure

To enable uploading a FastDoc batch with the Web Client upload service:

1. In Datacap, click the Administrator tab and then click Workflow
2. In the Parameters section, set the Program value to `uplBFcl.aspx`.
3. Click Apply.
4. On the scanning station, open the `dc2run.exe.config` file in a text editor.
5. Set the `StrictUpload` setting to False:

```
<setting name="StrictUpload" serializeAs="String">
<value>False</value>
</setting>
```

6. Set the FDPATHMASK setting. For example:

```
<setting name="FDPATHMASK" serializeAs="String">
<value>c:\datacap\FastDoc\batches\*\*\finished.xml</value>
</setting>
```

The service looks for any page files of the specified name. FastDoc saves a page file with the name finished.xml when the offline workflow is complete.

Parent topic: [Installing and configuring the Datacap Web Client upload service](#)

Using Datacap Web Client upload service with Datacap Desktop

This section provides information about how to create a working configuration for using Datacap Desktop, storing the scanned image files locally, and then uploading them automatically to the Datacap Server using the Datacap Web Client Upload Service.

For detailed procedure to install and configure the Datacap Web Client Upload Service, see [Installing and configuring the Datacap Web Client upload service](#)

Overview

One way to increase the performance of using Datacap Desktop (DcDesktop) is to configure it to store the scanned image files locally, the upload, instead of DcDesktop saving them as it goes over the WAN to the run-time batch directory. This approach is completely optional and DcDesktop can work without using this approach.

Note: This configuration adds complexity to the implementation, adds additional configuration to maintain, and adds more “active” Taskmaster components to each client PC. You might want to try the approach as it provides more benefits.

Configuration

This functionality involves at least three different "servers": Datacap Server, Datacap Web Server, at least one "client" PC running DcDesktop. There can be any number of client PCs running DcDesktop. All of these client PCs must run its own copy of the Datacap Web Client Upload Service. This is a "push" delivery design when the many client machines individually push their batch images to the Datacap Server.

There is work to be accomplished on both the central system and work to be accomplished on each Datacap client machine.

One-time procedure

1. On the network share, typically the Datacap share, create a shared network directory with a name something like "[servername]\Datacap\UploadClientConfig". This is where a special copy of the datacap.xml file resides that is leveraged by all client PCs that participate in this DcDesktop-Upload configuration.
2. Into this network shared directory "UploadClientConfig" copy the current "datacap.xml" file. This shared file contains UNC-based paths to the various TM applications (in a multi-server environment) and is

where the Datacap Application Manager on the client PCs usually reference. In this special situation, the "standard" datacap.xml file is not used because of the changes to be described next.

3. Text-edit the datacap.xml in the shared "UploadClientConfig" directory to change the UNC path reference to the new share directory described above ("[local drive letter]:\Datacap\[application name]").

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<datacap ver="8.0">
    <app name="DotScanUploader" ref="C:\Datacap\DotScanUploader "/>
</datacap>
```

4. The web client Admin configuration for the service feature to work requires a "Scan" task for DcDesktop. And an "Upload" task for the web client upload service.

Details for the scan task:

This is a Batch Creation task and uses the "DcDesktop" program. Make sure to select:

- o "Mode" to "Batch creation"
- o "Store" to "Station ID"

In the "Setup" configuration menu, set any Page file name. Enable the "Create batch directory" option. Set a Main task profile value. In the "DcDesktop" configuration area, set the appropriate option (ISSScan, TWAINScan, VScan, and so on.)

Details for the Upload task:

Next, configure the Upload task. The important items are:

- o The "UplBFcl.aspx" selection
- o The "Queue by" must be "Station" (or "Station ID" depending on the version)

The Setup configuration is as below. Enable the option "Create batch directory".

5. The Datacap Studio configuration for the sample project demonstrates what is required for the upload service feature to work.

Note: No configuration items are actually required for either the DcDesktop or client upload functionality.

6. Open the Admin database. View the Buttons table. Find the "Upload" row. If the "btn_FileName" column is blank, enter the value as "tmclient.exe". The required information is the value for that row in the "btn_ButtonNumber" column. Record this value for use in a later step.

On First Client machine

Perform the following steps:

1. Install TM Web Client Upload Service. For details, see [Installing and configuring the Datacap Web Client upload service](#)
2. Create a directory anywhere on the client PC where DotScan with temporarily save the scanned image files. In this example "C:\BatchesToUpload".
3. Create a directory under the local Datacap installation directory, that shows the name of the selected TM application. In this example "DotScanUploader". (i.e. "C:\Datacap\DotScanUploader")
4. Copy into this directory a copy of the selected TM application's .APP file (that is, copy \\TMserver1\Datacap\DotScanUploader\DotScanUploader.app to C:\Datacap\DotScanUploader\DotScanUploader.app)
5. Open the TM App Manager and on the Service tab browse to the shared directory described in the previous instructions to find the datacap.xml. This leverages the edited copy of the selected TM application's APP file in the "[servername]\Datacap\UploadClientConfig" directory.

Complete the configuration items on the Main tab, setting the "Batch folder" value to the local directory created in a previous step (that is, "C:\BatchesToUpload"). The remaining values must be UNC-based to

match the corresponding values in the primary .APP file.

On the Taskmaster tab check that the connection string values are NOT blank. If they are blank, that means that the encryption keys (dc_KTF.xml and Windows keystore) on the client machine don't match the keys on the TM Server. Update the encryption keys by using the dcskey.exe utility.

Important: When you close the TM App Manager, the "blank" connection strings overwrite the earlier entries, so a new copy of the .APP file should be created from the original and used again after correcting the encryption keys.

The Rulerunner tab requires no values for the upload service to work.

On the Custom values tab, in the Advanced values area, create a dc2run user log-in.

The Datacap App Manager configuration is now complete. Close and reopen the App Manager to confirm that the configuration was correctly saved.

6. Edit the file "dc2Run.exe.config" file. The critical configuration items are:
 - o "dc2Run_localhost_utility" value is your TM web server.
 - o "Application" value is the name of the TM application to upload.
 - o "User" is the name of the user that is used by the web client upload service. This is the name of the user that is included in the TM App Manager's Advanced custom variable (in this case, "dc2run.admin").
 - o "Station" is the station name under which the upload service is running.
 - o "ShortcutIndex" is the number from the "Upload" row of the Buttons table in the Admin database.
 - o "StrictUpload" is value "True" for uploading the documents that are saved after DotScan.
 - o "Worktimes" is value 00:01-23:59 for the entire day.
 - o "FDPathMask" must be blank.

```
<?xml version="1.0"?>
<configuration>
  <configSections>
    <sectionGroup name="applicationSettings"
type="System.Configuration.ApplicationSettingsGroup, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
      <section name="dc2Run.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false"/>
    </sectionGroup>
  </configSections>
  <applicationSettings>
    <dc2Run.Properties.Settings>
      <setting name="dc2Run_localhost_utility" serializeAs="String">
        <value>http://tmweb1/tmweb.net/Task/utility.asmx</value>
      </setting>
      <setting name="Application" serializeAs="String">
        <value>DotScanUploader</value>
      </setting>
      <setting name="User" serializeAs="String">
        <value>admin</value>
      </setting>
      <setting name="Delay" serializeAs="String">
        <value>3000</value>
      </setting>
      <setting name="RetryDelay" serializeAs="String">
        <value>15000</value>
      </setting>
      <setting name="Station" serializeAs="String">
        <value>1</value>
      </setting>
      <setting name="ShortcutIndex" serializeAs="String">
```

```

        <value>16</value>
    </setting>
    <setting name="WorkTimes" serializeAs="String">
        <value>00:01-23:59</value>
    </setting>
    <setting name="StrictUpload" serializeAs="String">
        <value>True</value>
    </setting>
    <setting name="FDPathMask" serializeAs="String">
        <value></value>
    </setting>
    </dc2Run.Properties.Settings>
</applicationSettings>
<startup><supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
</startup></configuration>

```

7. Open DcDesktop and create a batch. After creating the new batch, look in the local scan directory. There should be a batch sub-directory containing TIF files and an XML file. If not, back up and reconfigure.
8. Open the Event Viewer. Select the Windows Logs, Application. The select Clear Log...
9. Open the Windows Services and set the Log On account.
10. Start the Datacap Web Client Upload Service to perform a test upload. For details, see [Starting the Datacap Web Client upload service](#)
11. Refresh the Event Viewer. A successful upload is indicated by entries and the local batch directory is emptied.

On subsequent client machines

Complete the following, which includes references to instruction steps of the previous section; see details for each step there.

1. Install TM Web Client Upload Service. For details, see [Installing and configuring the Datacap Web Client upload service](#)
2. Create temp image folder.
3. Create selected TM application folder under [driveletter]:\Datacap
4. Copy the .APP file from the first client machine's folder listed in the previous step. The .APP need not be created again.
5. Open the Datacap App Manager and in the Service tab select the same datacap.xml file as in the first client machine. Verify the Datacap App Manager settings that appear preconfigured.
6. Copy the file "dc2Run.exe.config" file created in the previous section, step 7 to this client PC, [driveletter]:\Datacap\Taskmaster directory.
7. Perform the Steps 8 to 12 from the previous section to test the upload on this client machine.
Tip: You can also automate many of these steps with a batch file pointing to the appropriate files/directories to copy.

Parent topic: [Installing and configuring the Datacap Web Client upload service](#)

Starting the Datacap Web Client upload service

You configure the Datacap Web Client upload service differently depending on where the Scan Into is located.

About this task

You can configure the service to start automatically when the scan station is started.

Procedure

To start the Datacap Web Client upload service:

1. From the Windows Start menu on the scan station, click Control Panel > Administrative Tools > Services.
2. Right-click Datacap Datacap Web Client Service and select Properties.
3. Change the Startup type to Automatic. Then, click Apply.
4. Start the service:

Option	Description
If the Scan Into folder is on the scan station	<ol style="list-style-type: none">a. Click Start to start the service.b. After the service starts, click OK to close the Property dialog box. The service is now functional.c. Close the Services window.
If the Scan Into folder is not on the scan station	<ol style="list-style-type: none">a. Click the Log On tab and select This account.b. Locate or enter the domain name, user name, and password of the account that the Datacap Web Client Service is to use, and click Apply.c. Click OK to close the message box. Then, click Start to start the service. After the service starts, click OK to close the Property dialog box. The service is now functional.d. Close the Services window.

Parent topic: [Installing and configuring the Datacap Web Client upload service](#)

Viewing the Event Log

If you do not get the results you expect when you use the Datacap Web Client upload service, view the Application level log in the Event Viewer for the scan station. The log is where any issues or successfully completed tasks are logged.

Procedure

To view the Event Log:

1. From the Windows Start menu for the scan station, click Control Panel > Administrative Tools > Event Viewer.
2. Click Application to view a list of log entries. Log entries that are created by the Datacap Web Client Service show a Source name of Datacap Web Client Service.
3. Double-click a log entry to view it.

Parent topic: [Installing and configuring the Datacap Web Client upload service](#)

Disabling the Datacap Web Client upload service

You can disable the Datacap Web Client upload service if you are not using the service.

About this task

Procedure

To disable the Datacap Web Client Web Client upload service:

1. Log on to the scan station.
2. Click Start > Administrative Services > Services, then click Datacap Web Client Service.

3. Right-click on the highlighted Datacap Datacap Web Client Service service, and click Stop.

Parent topic: [Installing and configuring the Datacap Web Client upload service](#)

Starting Datacap Studio

This procedure provides instructions on how to start the Datacap Studio software component.

Before you begin

Before starting Datacap Studio, [start or ensure the Datacap Server Service is started](#).

Procedure

To start Datacap Studio:

1. From the developer workstation's Windows Start menu, select IBM Datacap Developer Tools>Datacap Studio. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.
2. Select the application that you want to work with, then click Next.
3. The Datacap Login window opens.
4. Enter a valid Datacap user ID, Password, and Station ID, then click Finish. The Datacap Studio main window opens with the Rulemanager tab displayed.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Installing and configuring Datacap Maintenance Manager

Datacap Maintenance Manager is a Datacap software component that sets up batch monitoring, status notification, and automatic deletion of completed batches.

You can use Datacap Maintenance Manager to monitor the status of your Datacap applications.

- Identify batches that meet certain criteria, such as batches that stopped running
- Change the status or order of batches in the job queue to correct problems that occur regularly. For example, reset a batch status from aborted to pending
- Delete batches or archive batches that completed processing, replaces AutoDelete
- Generate data for reporting with Datacap Report Viewer. For example, you can take a snapshot of the number of current users
- Notify an administrator that something went wrong with a background server, such as a Rulerunner server
- Send email notifications or take other actions.

Maintenance Manager includes the following components:

Maintenance Manager

A configuration utility for creating the required settings file and running Maintenance Manager rulesets manually.

Maintenance Manager actions

A library of actions that you can use to connect to a Datacap application and query the database for batch information. Modify information in the database, move or delete batches, and send notifications. You can use these actions in rulesets in an existing Datacap application or you can create a new application specifically for batch monitoring.

You can run Maintenance Manager manually by using Maintenance Manager or automatically at scheduled times by using the Windows Task Scheduler.

To install and configure Maintenance Manager, you must complete the following steps.

1. Create or ensure that an account exists for Maintenance Manager, and grant that account the appropriate sharing and security permissions on the Datacap Server.
 2. On the computer on which the Maintenance Manager process (NENU.exe) is run, add the Maintenance Manager domain/Windows account to either the Administrators or Backup Operators group.
 3. Ensure that the developer was granted sharing and security permissions on the Datacap Server by following the instructions in the section titled [Configure Datacap on Server](#).
 4. Install Maintenance Manager on the developer workstation.
 5. After you install and configure Maintenance Manager in a client/server environment, follow the instructions in [Creating a Maintenance Manager application](#) to develop your custom Maintenance Manager application and set up Windows Task Scheduler to run your Maintenance Manager application.
- [Creating or ensuring an account exists for Datacap Maintenance Manager](#)
Ensure a domain/Windows account exists for Datacap Maintenance Manager. If a domain/Windows account does not exist for Maintenance Manager, you must create one.
 - [Setting Datacap Maintenance Manager account permissions for sharing](#)
You set up the appropriate sharing permissions for the Datacap Maintenance Manager account on the Datacap Server shared c:\Datacap folder.
 - [Setting Datacap Maintenance Manager account security permissions on the Datacap folder](#)
You must set up the appropriate security for the Datacap Maintenance Manager account on the Datacap Server shared c:\Datacap folder.
 - [Setting Datacap Maintenance Manager account security permissions on the Datacap\RRS folder](#)
You must set up the appropriate security for the Datacap Maintenance Manager account on the Datacap Server shared c:\Datacap\RRS folder.
 - [Installing the developer workstation software components](#)
Run the Datacap installation program wizard on the workstation of a developer to install the Datacap software components.
 - [Importing encryption keys to Datacap computers](#)
In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Creating or ensuring an account exists for Datacap Maintenance Manager

Ensure a domain/Windows account exists for Datacap Maintenance Manager. If a domain/Windows account does not exist for Maintenance Manager, you must create one.

About this task

Datacap does not require that you set up a unique domain/Windows account for Datacap Maintenance Manager. Maintenance Manager can use any Windows account if account can be set up with the appropriate sharing and security permissions.

Parent topic: [Installing and configuring Datacap Maintenance Manager](#)

Setting Datacap Maintenance Manager account permissions for sharing

You set up the appropriate sharing permissions for the Datacap Maintenance Manager account on the Datacap Server shared c:\Datacap folder.

About this task

These instructions are intended for server environments that are running on supported versions of Microsoft Windows. Other accounts were granted sharing permissions during the installation and configuration of Datacap.

Procedure

To set Maintenance Manager account permissions for sharing:

1. On the Datacap Server computer, start Windows Explorer.
2. Go to c:\Datacap, right-click on the c:\Datacap folder and select Properties.
3. On the Properties dialog, click the Sharing tab. The folder must already be shared with the Share name of Datacap.
4. On the Sharing tab, click Advanced Sharing.
5. If the User Account Control window displays, click Yes.
6. On the Advanced Sharing dialog, click Permissions.
7. On the Permissions for, verify the domain/Windows user ID of Maintenance Manager is set to allow Full Control.

Parent topic: [Installing and configuring Datacap Maintenance Manager](#)

Setting Datacap Maintenance Manager account security permissions on the Datacap folder

You must set up the appropriate security for the Datacap Maintenance Manager account on the Datacap Server shared c:\Datacap folder.

About this task

These instructions are intended for server environments that are running on supported versions of Microsoft Windows. Other accounts were granted security permissions during the installation and configuration of Datacap.

Procedure

To set Maintenance Manager account security permissions on the Datacap folder:

1. On the Datacap Server computer, start Windows Explorer.
2. Go to c:\Datacap, right-click on the c:\Datacap folder and select Properties.
3. On the Properties dialog, click the Security tab.
4. Verify the Domain/Windows user ID of Maintenance Manager is set to allow Read & Execute.

Parent topic: [Installing and configuring Datacap Maintenance Manager](#)

Setting Datacap Maintenance Manager account security permissions on the Datacap\RRS folder

You must set up the appropriate security for the Datacap Maintenance Manager account on the Datacap Server shared c:\Datacap\RRS folder.

About this task

These instructions are intended for server environments that are running on supported versions of Microsoft Windows. Other accounts were granted security permissions during the installation and configuration of Datacap.

Procedure

To set Maintenance Manager account security permissions:

1. On the Datacap Server, start Windows Explorer.
2. Go to c:\Datacap\RRS, right-click on the c:\Datacap\RRS folder and select Properties.
3. On the Properties dialog, click the Security tab.
4. Verify the Domain/Windows user ID of Maintenance Manager is set to allow Read & Execute.

Parent topic: [Installing and configuring Datacap Maintenance Manager](#)

Installing the developer workstation software components

Run the Datacap installation program wizard on the workstation of a developer to install the Datacap software components.

About this task

The software components that you install include the Datacap client and sample applications. Also, the separately licensed applications, and connectors for which you have a license, the Datacap Studio, FastDoc, and Maintenance Manager software components.

These instructions apply to workstations that run on Windows 7.

Procedure

To install the developer workstation components:

1. Put the installation package on your network, or insert the Datacap CD in the developer workstation CD/DVD drive. If the installation process does not start automatically, or if the package is on the network, open Windows Explorer, go to and double-click the Setup.exe. Click Yes at the User Account Control window.
2. Select the appropriate language and click OK. The language that you select is displayed on the installation program screens during the installation.
3. When more, redistributive software is required, the installation program wizard displays a list of the items that must be installed, click Install.
4. Click Next.
5. Click to accept the license agreement and click Next.
6. Select the Custom option and click Next.

7. Exclude all of the components except the Datacap Client and the separately licensed applications and connectors for which you have a license. Be sure the FastDoc, Datacap Studio, and Maintenance Manager components are included.
8. Click Next.
9. Click Install.
10. Click Finish.

Parent topic: [Installing and configuring Datacap Maintenance Manager](#)

Importing encryption keys to Datacap computers

In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.

Before you begin

You must generate the encryption keys in the keystore on a server on which the Datacap server software component is installed. You export the new keys to a key transport file.

Procedure

To import encryption keys to Datacap computers:

1. Find the dc_KTF.xml key transport file in the c:\Datacap\Taskmaster folder on the Datacap server where you generated and exported the encryption keys.
2. Copy the dc_KTF.xml key transport file to the appropriate folder on the computer where you installed the new component. The encryption keys are automatically applied to the keystore the next time you start the Datacap component.

Table 1. Encryption key folder locations by component

Component	Folder
Datacap Server	C:\Datacap\Taskmaster
Datacap applications	C:\Datacap\Taskmaster
Datacap Desktop	C:\Datacap\DcDesktop
Datacap Studio	C:\Datacap\DStudio
Datacap FastDoc	C:\Datacap\FastDoc
Datacap Report Viewer	C:\Datacap\RV2\bin
Datacap Web Client	C:\Datacap\tmweb.net\bin
Datacap Web Services	C:\Datacap\wTM\bin

Parent topic: [Installing and configuring Datacap Maintenance Manager](#)

Configuring authentication for Datacap

You can configure Datacap to use Datacap authentication or to use one of the external authentication methods.

About this task

The authentication system that you choose affects whether you set up and maintain individual users or groups in your Datacap applications. The following five authentication systems are supported:

- Datacap authentication (referred to as TMA)
 - Windows Active Directory (referred to as ADSI)
 - Windows Active Directory Lightweight Directory Services (referred to as ADLDS)
 - Lightweight Directory Access Protocol (referred to as LDAP)
 - Low-Level Lightweight Directory Access Protocol (referred to as LLDAP)
- Important: For Datacap Navigator authentication, you can use ADSI (password required), ADLDS, LDAP (password required), and LLDAP. Datacap Navigator does not support LDAP and ADSI authentication using Windows account credentials.

For a planning overview related to domains, Windows accounts, authentication systems, Datacap users, groups, stations, and databases, see [Planning your Datacap system](#).

- [Datacap Server service settings](#)
The Datacap Server runs as a Windows service that you can start, pause, or stop in the Datacap Server Manager window. When you open the Datacap Server Manager dialog window, you can also configure authentication, database access, batch processing, and the logging level.
- [TMA authentication system](#)
The Datacap TMA authentication system supports both user and group authentication. You can select the TMA option from the Datacap Server Manager list of authentication systems. When you select the TMA authentication system option, the group names, user names, and passwords that you set up in the Datacap applications are used for authentication. You can set up user authentication credentials for the Maintenance Manager application, Rulerunner service, Datacap Web Client Upload service, and Datacap web services.
- [Active Directory ADSI and LDAP authentication systems](#)
The Active Directory ADSI and LDAP systems authenticate at the group level. You can select the ADSI or LDAP authentication system option from the Datacap Server Manager list of authentication systems. When you select the ADSI or LDAP authentication option, the credentials from the Windows account are used for authentication. For Active Directory domains that have a mutual trust relationship, ADSI and LDAP systems support the authentication of users in multiple domains.
- [ADLDS and LLDAP authentication systems](#)
When the ADLDS or LLDAP authentication system is used, the user names and passwords that are entered on Datacap login windows or passed to Datacap by background services and processes are used for authentication.
- [LLLDAP group authentication](#)
In the Datacap Server Manager, you can select LLDAP as your group authentication system. When you select LLDAP as your group authentication system, you must set up the Authentication path template with customized directory configuration properties.
- [Configuring the Datacap Server service to use an external authentication system](#)
You can configure the Datacap Server service for external authentication. When you open the Datacap Server Manager, you can select the ADSI, ADLDS, LDAP, or LLDAP authentication method.
- [Authenticating Datacap Web Client users with IBM Security Access Manager](#)
Datacap Web can decode WebSEAL directed headers and use the information that is provided in the headers to authenticate the Datacap user. The extracted identity information is used to retrieve the LDAP group information for that user by using an LDAP directory bind. Complete the following steps to authenticate users who are authenticated in Security Access Manager.
- [Configuring single sign-on \(SSO\) for Datacap Navigator](#)
You can configure single sign-on (SSO) for Datacap Navigator by using IBM Security Access Manager, SPNEGO/Kerberos, or container-managed authentication.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Related concepts:

[Datacap Web Services authentication](#)

Related tasks:

[Planning your Datacap system](#)

Datacap Server service settings

The Datacap Server runs as a Windows service that you can start, pause, or stop in the Datacap Server Manager window. When you open the Datacap Server Manager dialog window, you can also configure authentication, database access, batch processing, and the logging level.

Datacap Server Manager

There are four tabs in the Datacap Server Manager dialog window from which you can configure the Datacap Server service.

Table 1. Datacap Server Manager configuration dialog window

Name	Description
Service tab	<p>The Service name of the Windows service is Datacap Server.</p> <p>The Status options are shown as Running, Paused, or Stopped.</p> <p>You can start, pause, or stop the Datacap server service by selecting the appropriate icon that is shown under the Status area.</p>
Datacap tab	<p>From the Queue by list, select job or task to determine whether queued batches are processed first by job or by task.</p> <p>Select the Save deleted batch info in debug table check box to ensure that information from deleted batches is stored in the debug table of the Engine database of the application.</p> <p>Select the Child job inherits priority check box to ensure the priority property of a child job, such as Fixup, is the same priority as its parent job.</p> <p>In the Advanced settings pane, you can configure the port connection, set the database command timeout, select the type of authentication, and configure the Batch naming template. For more information about Advanced settings, see <i>Advanced settings for the Datacap Server service</i>.</p>
File system access mask tab	<p>You can add, remove, and edit folders, files, and file categories to which the Datacap Server service is allowed or denied access.</p>
Logging tab	<p>The System event log records Windows events. Your placement of the slider in the Messaging Level area determines the volume and criticality of messages that are logged.</p> <p>In the Datacap log pane, you can select the type and volume of Datacap messages that are logged. The Number of Messages slider indicates the volume and criticality of messages that are logged. Select the Output to file check box to specify the file name and location of the Datacap log file.</p>

- [Advanced settings for the Datacap Server service](#)

In the Advanced settings pane of the Datacap Server Manager, you can configure the Datacap Server

service connections port and set the database command timeout. You can also select your authentication method and configure the batch naming template.

Parent topic: [Configuring authentication for Datacap](#)

Advanced settings for the Datacap Server service

In the Advanced settings pane of the Datacap Server Manager, you can configure the Datacap Server service connections port and set the database command timeout. You can also select your authentication method and configure the batch naming template.

Datacap Server Manager Advanced settings

Table 1. Advanced settings

Name	Description
Accept connections on port Field	Designates the TCP/IP port from which the Datacap Server service accepts connections.
Database command timeout Field	The timeout value for ADO and OLEDB calls. The value that you enter represents the number of seconds the Datacap Server service waits for a connection before the call attempt is terminated.
Authentication system List	Select TMA for the Datacap authentication system or select one of the external authentication systems. For more information about configuring for external authentication, see <i>Configuring the Datacap server service to use an external authentication system</i> .
Authentication path template Field	<p>Enter the string in the Authentication path template field that is appropriate for your authentication type. You can modify the default path that is based on the specific address requirements in your environment.</p> <p>For ADSI and LDAP, the user password verification can be enabled by adding a value in the authentication template, <code>?password:enabled</code>. For ADSI, the template is <code>winNT://<%domain%>/<%user%>?password:enabled</code> and for LDAP the template is <code>LDAP://<%domain%>.com?password:enabled</code>.</p>
Batch naming template Field	<p>The batch ID generation is defined in the Batch naming template setting. The batch ID can be up to 50 characters long and can contain a mix of string constants and variable fields. A typical batch ID combines a date with a sequential number.</p> <p>The default value for the Batch naming template setting is <code>[Y][m][d].[n:6]</code>. For example, if the date of a batch generation is December 16, 2013, the first batch ID is <code>20131216.000000</code> and the second batch ID is <code>20131216.000001</code>.</p> <p>The batch ID can contain other text. For example, the Batch naming template setting, <code>a[Y]bc[m]d[d]_[N:6]</code>, generates the batch ID, <code>a2013bc12d16_000000</code>.</p>
Sample batch Field	When you enter the Batch naming template setting, the Sample batch field automatically shows the structure of the batch ID that is generated.

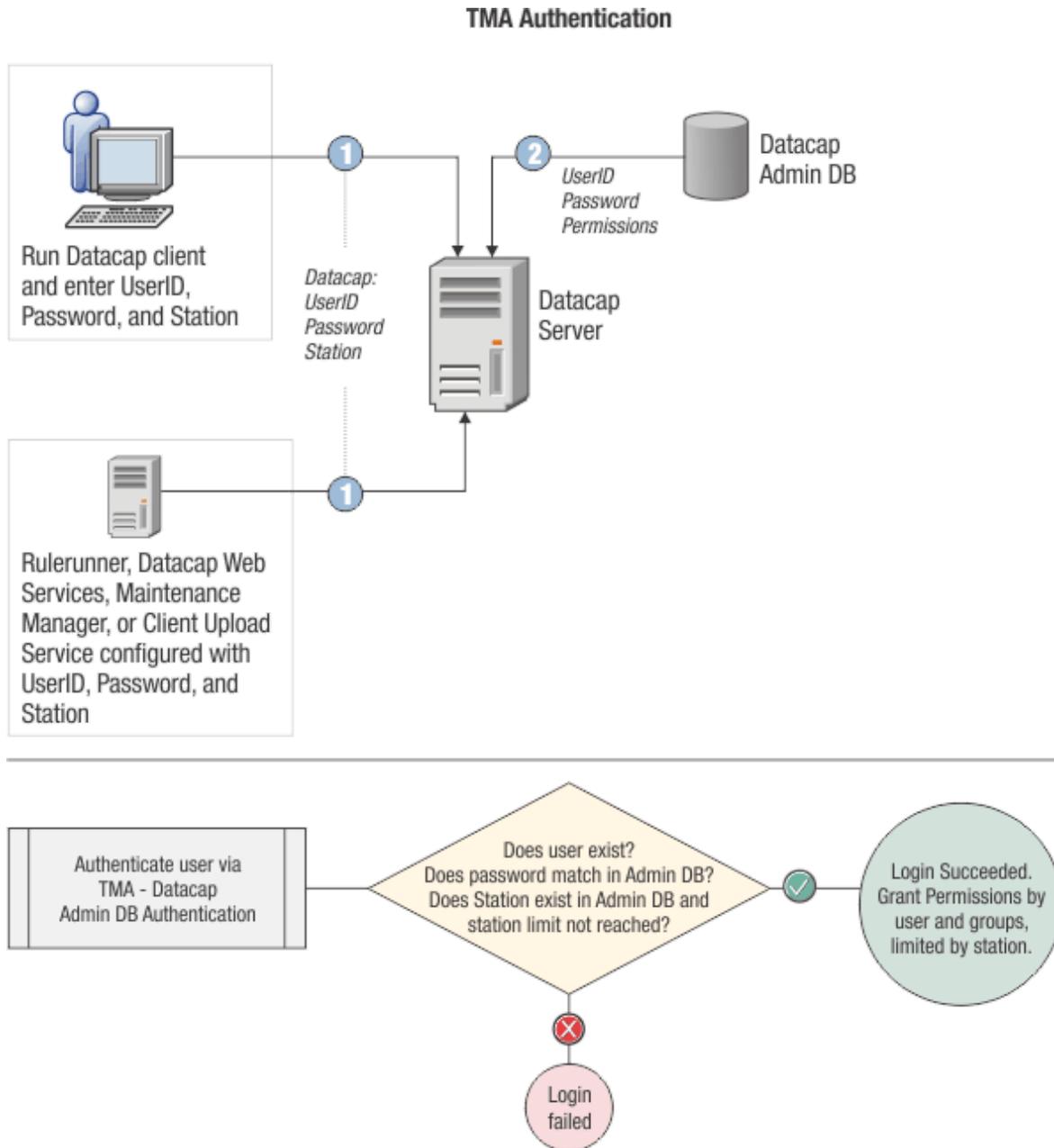
Parent topic: [Datacap Server service settings](#)

Related tasks:

[Configuring the Datacap Server service to use an external authentication system](#)

TMA authentication system

The Datacap TMA authentication system supports both user and group authentication. You can select the TMA option from the Datacap Server Manager list of authentication systems. When you select the TMA authentication system option, the group names, user names, and passwords that you set up in the Datacap applications are used for authentication. You can set up user authentication credentials for the Maintenance Manager application, Rulerunner service, Datacap Web Client Upload service, and Datacap web services.



TMA Datacap Server Service

In Datacap Server Manager, set the Authentication system to TMA.

TMA Datacap users, groups, stations in each application

Add Datacap users to your application. The name and password that you specify when you create the Datacap user are the credentials that the user and background service or process uses when logging in to Datacap.

Datacap groups are optional. Add Datacap groups to your application when you want to manage permissions at the group level. You can add groups to Datacap for users, automatic users, and background services and processes. The Datacap group name can be any name that you want. There is no need to create Datacap groups for Datacap server service, Datacap Web Services, or application pools.

You can add Datacap users to one or more Datacap group. You must be a member of a group to modify membership of that group.

Add Datacap stations to your application with the appropriate permissions. Station names can be any name that you want.

TMA Maintenance Manager

The Windows Scheduler runs the Maintenance Manager application automatically and the Maintenance Manager application supplies its credentials automatically. When a Maintenance Manager rule set is added to a Datacap application, the rules supply credentials automatically.

- Add a Datacap user and password to your application for Maintenance Manager or use an existing Datacap user with appropriate permissions.
- Add a Datacap station to your application for Maintenance Manager and assign appropriate permissions or use an existing Datacap station with appropriate permissions.
- To set up Maintenance Manager credentials for TMA, add a custom name and value pair in the Application Manager Advanced values fields to hold the password for the Maintenance Manager Datacap user.
 - Value name: Create a value name that is recognizable or can be associated with the Maintenance Manager user password, such as `Maintenance ManagerPassword`.
 - Value: Enter the Datacap password of the Datacap Maintenance Manager user.
- In the Maintenance Manager application, set the parameter of the SetUser action to the Datacap user name.
- In the Maintenance Manager application, set the SetPassword action to use the APPVAR smart parameter to retrieve the value of `Maintenance ManagerPassword` from the Datacap application service.
- In the Maintenance Manager application, use the SetStation action to provide the Datacap station name.
- In Windows Scheduler, set the account in Security Options to the Windows account used by Maintenance Manager to run with highest privileges.

TMA Rulerunner Service

The Rulerunner Service is a background service that supplies its credentials automatically.

- Add at least one Datacap user for Rulerunner to the Datacap application or use an existing Datacap user with appropriate permissions.
 - If one instance of Rulerunner is set up to process tasks from multiple applications, the same Datacap user name and password must be added to all applications.
 - If multiple instances of Rulerunner are set up, they can all use the same Datacap user.
- Add one Datacap station for each Rulerunner, create one Datacap station for the Rulerunner to share, or use an existing Datacap station with appropriate permissions.
- Set up the credentials in each Rulerunner Manager by selecting the Datacap Authentication option on the Rulerunner Login tab. Enter the Datacap user name, password, and station for this instance of Rulerunner.

TMA Datacap Web Client Upload Service

The Datacap Web Client upload service is a Windows service that supplies its credentials automatically.

- Add at least one Datacap user for the upload service to the Datacap application or use an existing Datacap user with appropriate permissions.
- Add at least one Datacap station for the upload service to the Datacap application or use an existing Datacap station with appropriate permissions.
- Set up the credentials by adding a name and value pair in the Application Manager Advanced values fields to hold the password of the Datacap upload service user.
 - Value name: Must be `dc2run.User`
 - Value: Enter the password of the Datacap upload service user.
- In the Datacap Web Client Upload configuration file, set the value of the `<setting name="User" node` to the Datacap Upload Service user.
- In the Datacap Web Client Upload configuration file, set the value of the `<setting name="Station" node` to the Datacap Upload Service station.

TMA Application Pools

Datacap uses application pools for Datacap Web Client, Report Viewer, and the Fingerprint Service. When Datacap Web Client and Report Viewer are installed on the same web server, they must use the same Windows account. When the Fingerprint Service is also installed on the same web server, it can use the same Windows account or a different one. The Windows account that is assigned to the application pool allows the application pool to function. When you assign the Windows account to the application pool, you provide the Windows credentials that the application pool uses.

There is no need to set up Datacap users, stations, or groups for application pools.

TMA Datacap web services (wTM)

Datacap web services supplies its credentials automatically.

- Add a Datacap user for wTM to the Datacap application or use an existing Datacap user with appropriate permissions.
- Add a Datacap station for wTM to the Datacap application or use an existing Datacap station with appropriate permissions.
- Set up the credentials by adding a name and value pair in the Application Manager General string values fields to hold the user name and station name. Add a name and value pair in the Advanced values fields for the user password.
 - Value name: `wTMUser`
 - Value: Set to the Datacap user name.
 - Value name: `wTMStation`
 - Value: Set to the Datacap station name.
 - Value name: `wTMPassword`
 - Value: Set to the Datacap user password.

Parent topic: [Configuring authentication for Datacap](#)

Active Directory ADSI and LDAP authentication systems

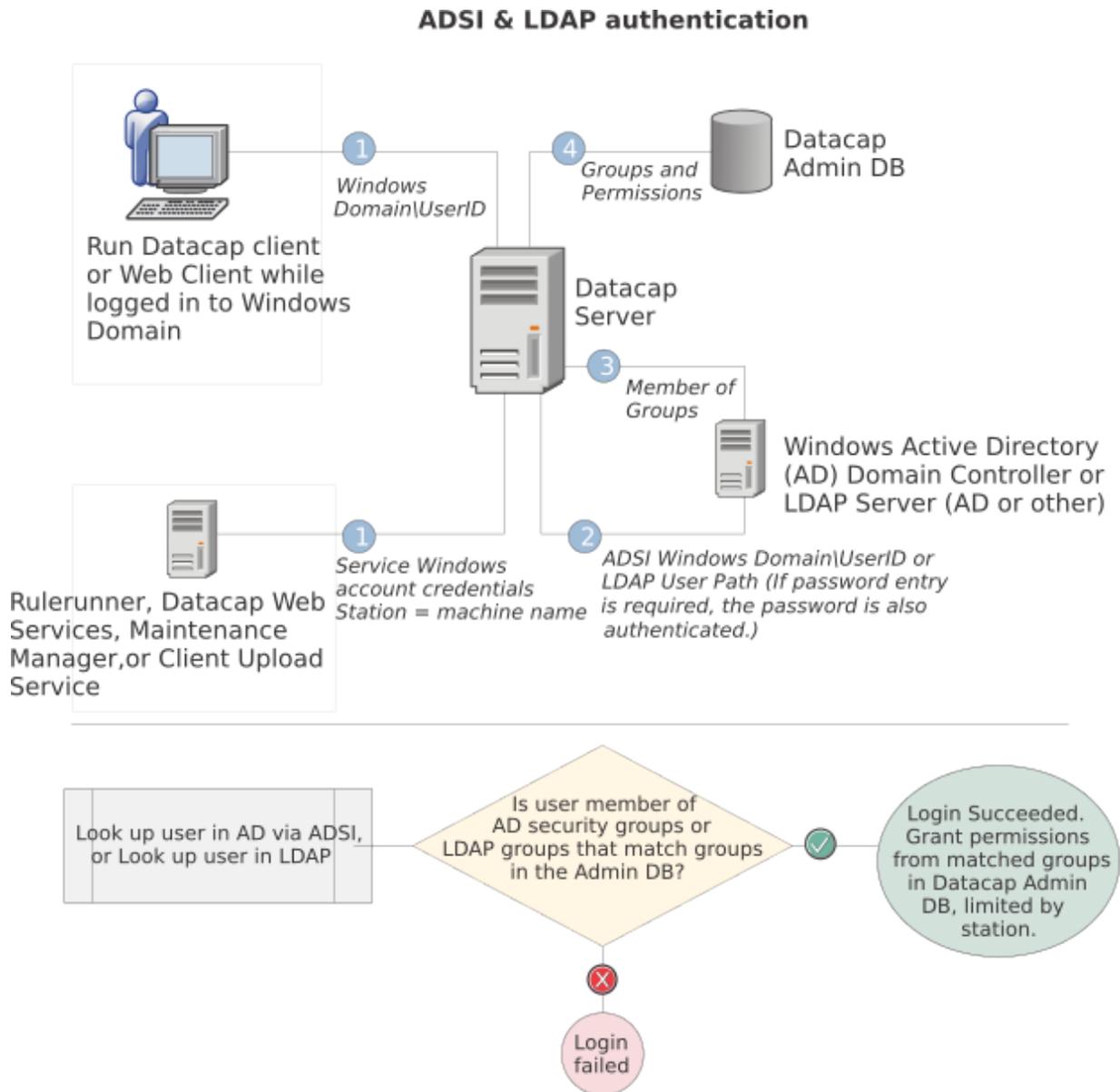
The Active Directory ADSI and LDAP systems authenticate at the group level. You can select the ADSI or LDAP authentication system option from the Datacap Server Manager list of authentication systems. When you select the ADSI or LDAP authentication option, the credentials from the Windows account are used for authentication. For Active Directory domains that have a mutual trust relationship, ADSI and LDAP systems support the authentication of users in multiple domains.

When you log into a Datacap application, you enter a password only if the authentication template string requires password entry. For information about authentication template strings, see [Configuring the Datacap Server service to use an external authentication system](#).

Active Directory ADSI or LDAP authentication in Datacap

Active Directory is referred to as ADSI in Datacap. You must ensure that the following tasks are completed when you are using the ADSI or LDAP authentication system.

- Appropriate security groups in Active Directory are created.
- Windows accounts are created for Datacap users, background services and processes, and application pools.
- The Windows accounts for Datacap are added to the appropriate Active Directory security groups.



ADSI or LDAP Datacap Server Service

In Datacap Server Manager, set the Authentication system to ADSI or LDAP.

ADSI or LDAP Datacap groups and stations

Depending on the number of ADSI or LDAP security groups you created, add corresponding groups to your Datacap application and assign Datacap permissions to each group. The Datacap group name must be in the following format:

- Active Directory security group name
- A dot
- Short domain name (domain without top level)

For example, if the Active Directory security group name is `TMUsers` and the full domain name is `domain02.com`, then the Datacap Group name must be `TMUsers.domain02`.

There is no need to create Datacap groups for the Datacap server service or for the Datacap Web Client, Report Viewer, and Fingerprint service application pools.

Add Datacap stations to your application with the appropriate permissions. Users of interactive Datacap software components enter station names manually so the station names for these users do not need to match their machine names.

For Maintenance Manager, Rulerunner, Datacap Web Services, and the Datacap Web Client Upload service, the machine names are provided automatically as the station name. These machine names must be added to your Datacap application as station names. Station names are case-sensitive.

When you are using ADSI or LDAP, authentication is performed at the group level and there is no need to add Datacap users to your Datacap applications.

ADSI or LDAP Datacap users

If password entry is not required, the Windows account of the user, background service, or background process is used for authentication. The following items apply when password entry is not required:

- Users that log in to interactive Datacap software components must enter a user name and station name. The user must not enter a password even though the Windows account information is used for authentication.
- Background services or processes must leave the user name, password, and station name blank because the Windows account information is used for authentication and the machine name is used as the station name.

For ADSI authentication, Internet Explorer users might need to add the Datacap Web Client `tmweb.net` address as a trusted site in their internet security options (for example, add `http://WebServerName`).

ADSI or LDAP Datacap Studio users

If password entry is not required, users must select the NT Authentication check box to log into Datacap Studio. Otherwise, if password entry is required, users must enter their full credentials.

ADSI or LDAP Maintenance Manager

The Windows Scheduler runs the Maintenance Manager application automatically. The Windows account that is used by the Maintenance Manager application and the computer name is used for authentication.

- Add a Datacap station to your application for Maintenance Manager that has the same name as the machine name and assign appropriate permissions.
- The Windows domain and user name must be used for SetUser to configure Maintenance Manager to authenticate to the Datacap server service.

- In Windows Scheduler, set the account in Security Options to the Windows account that is used by Maintenance Manager to run with highest privileges.

ADSI or LDAP Rulerunner Service

The Rulerunner Service is a background service that supplies its credentials automatically.

- Add a Datacap station to your application for each Rulerunner server and assign appropriate permissions. The station name in Datacap is case-sensitive and must match the machine name because it is maintained in the domain controller.
- If password entry is not required, set up the credentials in each Rulerunner Manager by selecting the Windows Authentication option on the Rulerunner Login tab. Otherwise, if password entry is required, users enter their full credentials.

ADSI or LDAP Datacap Web Client Upload Service

The Datacap Web Client upload service is a Windows service that supplies its credentials automatically.

- Add a Datacap station for the upload service to the Datacap application and assign appropriate permissions.
- Set up a blank password to be used by the upload service by adding a name and value pair in the Application Manager Advanced values fields.
 - Value name: Must be `dc2run.User`
 - Value: Leave this field blank.
- In the Datacap Web Client Upload configuration file, set the value of the `<setting name="User" node` to the domain and Windows account (for example `DOMAIN\UserID`) of the Datacap Upload Service user.
- In the Datacap Web Client Upload configuration file, set the value of the `<setting name="Station" node` to the Datacap Upload Service station.

ADSI or LDAP Application Pools

Datacap uses application pools for Datacap Web, Report Viewer, and the Fingerprint Service. When Datacap Web and Report Viewer are installed on the same web server, they must use the same Windows account. When the Fingerprint Service is also installed on the same web server, it can use the same Windows account or a different one. The Windows account that is assigned to the application pool allows the application pool to function. When you assign the Windows account to the application pool, you provide the Windows credentials that the application pool uses.

There is no need to set up ADSI or LDAP groups or Datacap users, stations, or groups for application pools.

ADSI or LDAP Datacap stations

Add a Datacap station to your application for wTM with the same name as the machine name and assign appropriate permissions.

Set up credentials as indicated by the following table:

Value name	Value	Credentials location
wTMUser	Leave this field blank.	Application Manager General string values fields
wTMStation	Set to the Datacap station name.	Application Manager General string values fields
wTMPassword	Leave this field blank.	Application Manager Advanced values fields

When you log into the station, you are prompted for credentials if password entry is required.

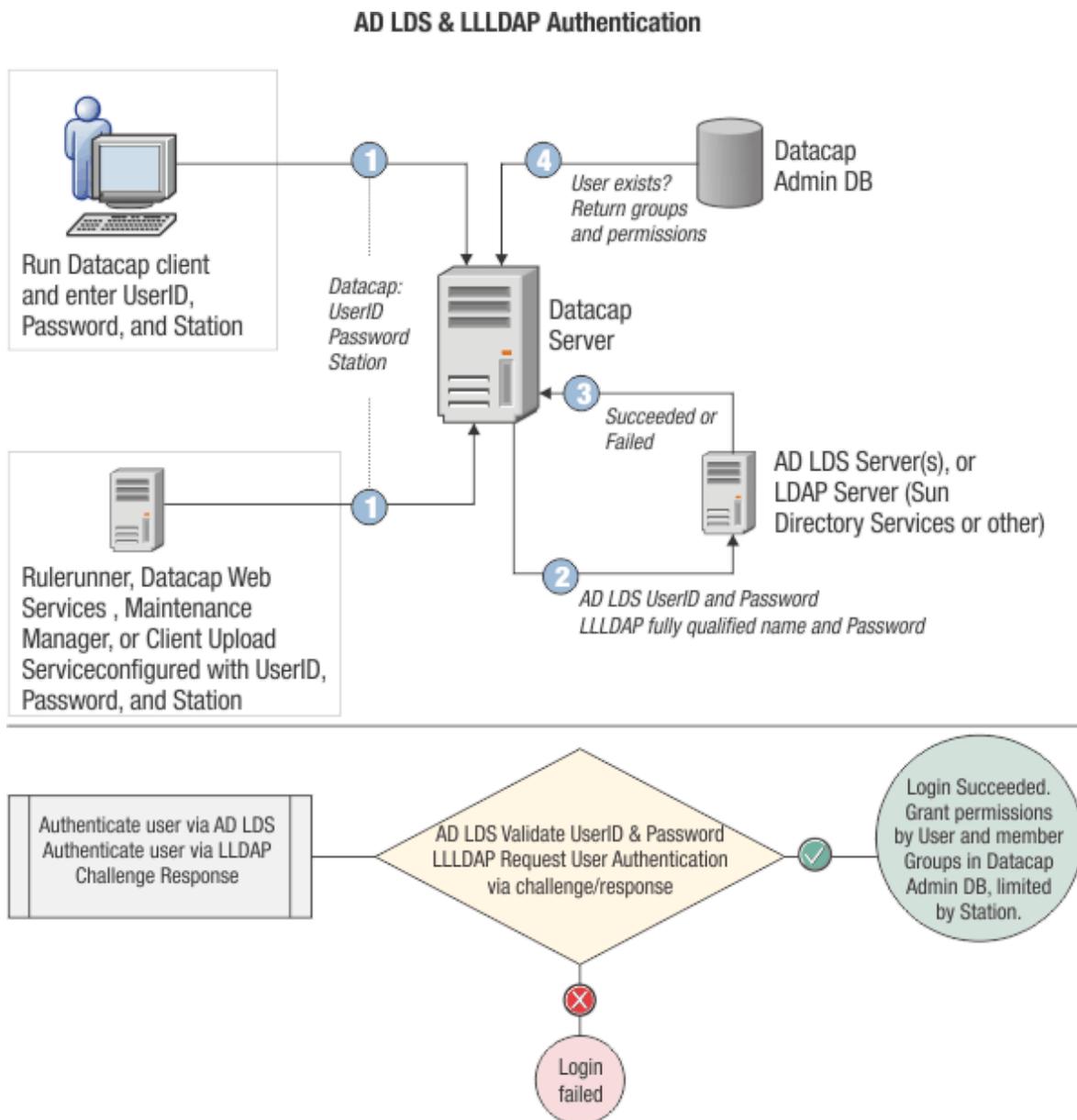
ADLDS and LLDAP authentication systems

When the ADLDS or LLDAP authentication system is used, the user names and passwords that are entered on Datacap login windows or passed to Datacap by background services and processes are used for authentication.

ADLDS or LLDAP authentication in Datacap

You must create accounts in ADLDS or LLDAP for Datacap users, background services, and background processes. Users must enter the user names and passwords of these accounts on the Datacap login windows. The background services and processes supply the account user names and passwords automatically.

You must also set up the same user names in your Datacap application. The user name and password is authenticated against your directory server. When the directory server authenticates successfully, the user name must match the user name in Datacap Administration Database.



ADLDS or LLDAP Datacap Server Service

In Datacap Server Manager, set the Authentication system to ADLDS or LLDAP as appropriate.

ADLDS or LLDAP Datacap users, groups, stations

Add Datacap users to your application with the same names and passwords that were set up in the ADLDS or LLDAP authentication system. The user or background service or process uses these credentials to log int to Datacap.

Datacap groups are optional. Add Datacap groups to your application when you want to manage permissions at the group level in addition to or instead of managing individual permissions. You can add groups to Datacap for users, automatic users, and background services and processes. The Datacap group name can be any name that you want. There is no need to create Datacap groups for the Datacap Server Service, Datacap Web Services, or application pools.

When appropriate, add Datacap users to one or more Datacap group or groups.

Add Datacap stations to your application with the appropriate permissions. Station names can be any name that you want.

ADLDS or LLDAP Maintenance Manager

The Windows Scheduler runs the Maintenance Manager application automatically and the Maintenance Manager application supplies its credentials automatically. When a Maintenance Manager rule set is added to a Datacap application, the rules supply credentials automatically.

- Add a Datacap user and password to your application for Maintenance Manager, or use an existing Datacap user with appropriate permissions. The user name and password must match a user name and password that is set up in the ADLDS or LLDAP authentication system.
- Add a Datacap station to your application for Maintenance Manager and assign appropriate permissions, or use an existing Datacap station with appropriate permissions.
- Set up Maintenance Manager credentials when you are using ADLDS or LLDAP. In Datacap Application Manager, select the application.
 - Add a General string name-value pair on the Custom values tab to hold the user name of the Maintenance Manager user as found in Datacap.
 - Add an Advanced values name-value pair on the Custom values tab to hold the password for the Maintenance Manager Datacap user.
 - Add one General string name-value pair on the Custom values tab to hold the Maintenance Manager station name as found in Datacap.
- In the Maintenance Manager application, set the SetUser action to use the APPVAR smart parameter to retrieve the value of the Maintenance Manager user name from Datacap application service.
- In the Maintenance Manager application, set the SetPassword action to use the APPVAR smart parameter to retrieve the value of the Maintenance Manager password from Datacap application service.
- In the Maintenance Manager application, set the SetStation action to use the APPVAR smart parameter to retrieve the value of the Maintenance Manager station name from Datacap application service.
- In Windows Scheduler, set the account in Security Options to the Windows account used by Maintenance Manager to run with highest privileges.

ADLDS or LLDAP Rulerunner Service

The Rulerunner Service is a background service that supplies its credentials automatically

- Add at least one Datacap user for Rulerunner to the Datacap application, or use an existing Datacap user with appropriate permissions. The user name and password must match a user name and password that is set up in the ADLDS or LLDAP authentication system. If one instance of Rulerunner is set up to process tasks from multiple applications, the same Datacap user name and password must be added to all of the applications. If multiple instances of Rulerunner are set up, they can all use the same Datacap user.
- Add one Datacap station for each Rulerunner, or create one Datacap station for the Rulerunners to share, or use an existing Datacap station with appropriate permissions.
- Set up the credentials that are used by Rulerunner when you are using ADLDS or LLDAP. In each Rulerunner Manager, select the Datacap Authentication option on the Rulerunner Login tab. Enter the Datacap user name, password, and station to be used for the instance of Rulerunner.

ADLDS or LLDAP Datacap Web Client Upload Service

The Datacap Client Upload Service is a Windows service that supplies its credentials automatically.

- Add at least one Datacap user for the Upload Service to the Datacap application, or use an existing Datacap user with appropriate permissions. The user name and password must match a user name and password that is set up in the ADLDS or LLDAP authentication system.
- Add at least one Datacap station for the Upload Service to the Datacap application, or use an existing Datacap station with appropriate permissions.
- Set up the credentials that are used by the Upload Service when you are using ADLDS or LLDAP. In Datacap Application Manager, select the application and add an Advanced values name and value pair on the Custom values tab to hold the password for the Upload Service Datacap user.
 - Value name – Must be `dc2run.User`
 - Value – Enter the password of the Datacap Upload Service user.
- In the Datacap Web Client Upload configuration file, set the value of the `<setting name="User"` node to the Datacap Upload Service user.
- In the Web Client Upload configuration file, set the value of the `<setting name="Station"` node to the Datacap Upload Service station.

ADLDS or LLDAP Application Pools

Datacap uses application pools for Datacap Web Client, Report Viewer, and the Fingerprint Service. When Datacap Web Client and Report Viewer are installed on the same web server, they must use the same Windows account. When the Fingerprint Service is also installed on the same web server, it can use the same Windows account or a different one. The Windows account that is assigned to the application pool allows the application pool to function. When you assign the Windows account to the application pool, you provide the Windows credentials that the application pool uses.

There is no need to set up Datacap users, stations, or groups for application pools.

ADLDS or LLDAP Datacap Web Services

Datacap Web Services supplies its credentials automatically.

- Add a Datacap user for Datacap Web Services to the Datacap application, or use an existing Datacap user with appropriate permissions. The user name and password must match a user name and password that is set up in the ADLDS or LLDAP authentication system.
- Add a Datacap station for Datacap Web Services to the Datacap application, or use an existing Datacap station with appropriate permissions.
- Set up the credentials that are used by Datacap Web Services when you are using ADLDS or LLDAP. In Datacap Application Manager, select the application.
 - Add a General string value name-value pair on the Custom values tab to hold the user name.

- Value name – `wTMUser`
 - Value – Enter the user name
- Add a General string value name-value pair on the Custom values tab to hold the station name.
 - Value name – `wTMStation`
 - Value – Enter the station name
- Enter an Advanced values name-value pair on the Custom values tab to hold the user password.
 - Value name – `wTMPassword`
 - Value – Enter the password

Parent topic: [Configuring authentication for Datacap](#)

LLLDAP group authentication

In the Datacap Server Manager, you can select LLLDAP as your group authentication system. When you select LLLDAP as your group authentication system, you must set up the Authentication path template with customized directory configuration properties.

LLLDAP authentication path template with customized directory configuration properties

The Authentication path template for LLLDAP group authentication must start with the `host:port` parameters, where `host` is the LDAP server name and `port` is the port number.

The `Bind User` and `Bind Password` can be encrypted by setting custom values the Application Manager. Values that are specified in the Advanced values field in the Custom values tab are encrypted. You must specify the application and the Value name in the Datacap Server Manager Authentication path template. For example, a password that is stored with the Value name, `MyBindPassword`, in the APT application Advanced values field, can be retrieved by the LLLDAP authenticator by specifying `APT, MyBindPassword` in the Authentication path template.

List of LLLDAP Directory Configuration Properties

Table 1. List of Directory Configuration Properties

Property Name	Description
<i>GroupBaseDn</i>	Group Base Domain Name The base domain name for searching for groups in the directory server.
<i>GroupSearchFilter</i>	Group Search Filter Specifies the search filter for groups, such as <code>(&(objectclass=group)(cn=<%user%>))</code> , where <code>cn</code> serves as the short name. <i>GroupSearchFilter</i> and <i>GroupDisplayNameAttribute</i> must use the same LDAP attribute
<i>GroupShortNameAtr</i>	Group Short Name Attribute Defines the directory server attribute to be used as the short name for a group.

Property Name	Description
<i>GroupDisplayNameAttr</i>	<p>Group Display Name Attribute</p> <p>Specifies the display name for a Group object that is generated by the authentication provider. The default property value is dependent on the authentication provider and is specified by the provider's configuration.</p>
<i>GroupMembershipSearchFilter</i>	<p>Group Membership Search Filter</p> <p>The search filter for group membership queries.</p>
<i>GroupNestedSearch</i>	<p>Group Nested Search Filter</p> <p>Include nested groups for discovering group membership. Valid values are On and Off. The default value is Off.</p>
<i>UserBaseDn</i>	<p>User Base Domain Name</p> <p>The base domain name for searching for users in the directory server.</p>
<i>UserSearchFilter</i>	<p>User Search Filter</p> <p>Specifies search filter for users, such as (&(objectclass=user)(samAccountName=<%user%>)), where samAccountName serves as the short name.</p>
<i>UserShortNameAttr</i>	<p>User Short Name Attribute</p> <p>Defines the directory server attribute to be used as the short name for a user.</p>
<i>UserDisplayNameAttr</i>	<p>User Display Name Attribute</p> <p>Specifies the display name for a User object that is generated by the authentication provider. The default property value is dependent on the authentication provider and is specified by the provider's configuration.</p>
<i>BindUser</i>	<p>Bind User</p> <p>The user name for authenticating the users. This user must have permission to search the area of LDAP where users are located. This user must have permission to search outside of the user's groups and authorization. The login fails if the application user cannot bind. The bind user can be specified in plain text or can be encrypted by using the Advanced values field in the Custom values tab of the Application Manager. When the bind user is stored in the Application Manager, you must specify the application name and the Value name as Application,<value name>. For example, enter APT,MyBindUserValue, where APT is the Datacap application name and MyBindUser is the Value name in the Custom values field. You must specify only the Value name. Do not specify the complete @APPVAR path in the Advanced values field</p>

Property Name	Description
<i>BindPw</i>	<p>Bind Password</p> <p>The password for the <i>Bind User</i>. The bind password can be specified in plain text or can be encrypted by using the Advanced values field in the Custom values tab of the Application Manager. When the bind password is stored in the Application Manager, you must specify the application name and Value name as <code>Application,<value name></code>. For example, enter <code>APT,MyPasswordValue</code> where <code>APT</code> is the Datacap application name and <code>MyBindPassword</code> is the Value name in the Advanced values field. You must specify only the value name. Do not specify the complete <code>@APPVAR</code> path in the Advanced values field</p>
<i>AuthAttribute</i>	<p>Authorization Attribute</p> <p>The <i>AuthAttribute</i> value tells Datacap server to authenticate a user with an additional attribute value, such as <code>AuthAttribute:carLicense=1234</code>. This value is optional.</p>

Example 1 - LLDAP group authentication path template

The following is an example of an Authentication path template with customized directory configuration properties.

```
MyServer:389/
BindUser:cn=admin,dc=mydomain,dc=com?BindPw:APT,MyBindPassword?
UserBaseDn:ou=people, dc=mydomain,dc=com?UserSearchFilter:(
&(objectClass=inetOrgPerson)(cn=<%user
%>)?UserShortNameAttr:cn?UserDisplayNameAttr:uid?GroupBaseDn:
ou=groups,dc=mydomain,dc=com?
GroupSearchFilter:(
&(objectClass=groupOfNames)?GroupShortNameAttr:cn?GroupNestedSearch:on?
GroupDisplayNameAttr:cn?GroupMembershipSearchFilter:(
&(objectClass=groupOfNames)(member=<%user %>))
```

Example 2 - LLDAP group authentication path template

The following is an example of an Authentication path template to authenticate a user with the `sAMAccountName` attribute.

```
Server:389/
BindUser:cn=admin,dc=mydomain,dc=com?BindPw:BindPw:APT,MyBindPassword?
UserBaseDn:DC=mycomain,DC=com?UserSearchFilter:(
&(objectClass=user)(sAMAccountName=<%user%>)?
UserShortNameAttr:cn?UserDisplayNameAttr:uid?GroupBaseDn:
DC=mydomain,DC=com?GroupSearchFilter:(&(objectClass=group))?
GroupShortNameAttr:cn?GroupNestedSearch:on?GroupDisplayNameAttr:cn?
GroupMembershipSearchFilter:(&(objectClass=group)(member=<%user%>))
```

Parent topic: [Configuring authentication for Datacap](#)

Configuring the Datacap Server service to use an external authentication system

You can configure the Datacap Server service for external authentication. When you open the Datacap Server Manager, you can select the ADSI, ADLDS, LDAP, or LLDAP authentication method.

About this task

In a client/server environment, you can configure Datacap Server service to use an external authentication provider, rather than using the default Datacap authentication (TMA) option.

Procedure

To configure Datacap Server service to use external authentication:

1. With the Datacap Server Manager window open, stop the service.
2. Click the Datacap tab and click Show advanced.
3. In the Authentication system field, select the type of authentication that you want to use.

Option	Description
ADSI authentication	Select this option if you are running Windows Active Directory for authentication. The Active Directory server must be part of the domain in which the Datacap computers are located. Select this option when you do not want all users that are logged in to a Windows domain to retype their passwords when they log in to Datacap.
ADLDS authentication	Select this option if you are running Windows Active Directory Lightweight Directory Service for authentication. The AD LDS server can be outside of the domain in which the Datacap computers are located.
LLDAP authentication	Select this option, rather than LDAP, when you are using authentication providers other than Windows Active Directory for authentication. Select this option when you are using providers such as Oracle Directory Server Enterprise Edition (formerly known as Sun Directory Server Enterprise Edition). The LLDAP server can be outside of the domain in which the Datacap computers are located.
LDAP authentication	Select this option only if you require that your LDAP server is accessed by using the Distinguished Name (DN) of a directory object and the complete LDAP path name. The LDAP server must be part of the domain in which the Datacap computers are located.

4. Enter the string in the Authentication path template field that is appropriate for your authentication type. You can modify the default path that is shown in the following table for each authentication option that is based on the specific address requirements in your environment. The Datacap Server service reads this path entry to determine the credentials for the particular account. For example, when you configure Active Directory authentication, Datacap looks up the user in Active Directory by using syntax similar to WinNT://<%domain%>/<%user%>. The <%domain%> and <%user%> entries are automatically replaced with the login domain of the user account and the user account. Important: As shown in the table, for ADLDS and LLDAP options only, you must enter actual values in the template path for %server%, %port%, and %domain%. The <%user%> variable entry must be

retained as shown. As an example, you might enter a template path, server01:1099/uid=<%user%>,dc=domain02,dc=Com.

Option	Description
ADSI authentication	Select one of the following strings to enter based on whether you require password entry: <ul style="list-style-type: none">WinNT://<%domain%>/<%user%>WinNT://<%domain%>/<%user%>?password:enabled
AD LDS authentication	Enter %server%: %port%/uid=<%user%>,dc=%domain%,dc=Com
LLLDAP authentication	Enter %server%: %port%/uid=<%user%>,dc=%domain%,dc=Com
LDAP authentication	Select one of the following strings to enter based on whether you require password entry: <ul style="list-style-type: none">LDAP://<%domain%>.comLDAP://<%domain%>.com?password:enabled

5. Click Save to save your authentication path entry.
6. Click the Service tab and then click Start to start Datacap Server service.
7. Close the Datacap Server Manager window.

Parent topic: [Configuring authentication for Datacap](#)

Authenticating Datacap Web Client users with IBM Security Access Manager

Datacap Web can decode WebSEAL directed headers and use the information that is provided in the headers to authenticate the Datacap user. The extracted identity information is used to retrieve the LDAP group information for that user by using an LDAP directory bind. Complete the following steps to authenticate users who are authenticated in Security Access Manager.

Security Access Manager configuration

For Security Access Manager to be able pass the user information in the HTTP Header, you must configure a path between the WebSEAL server and the IIS server that hosts Datacap Web Server. The WebSEAL server uses the iv-user header value to send the user information.

To configure a path between WebSEAL and Datacap Web, type the following command on the WebSEAL side:

```
"server task default-webseald-server create -t tcp  
-h mywebserver.com -p 80 -c iv-user,iv-groups/junction"
```

Alternatively, you can use an SSL connection to create a secure connection between WebSEAL and Datacap Web by using the pdadmin command and typing the following on the WebSEAL side:

```
"server task default-webseald-server create -f -t  
ssl -h mywebserver.com -p 443 -c iv-user,iv-groups/junction"
```

where the SSL connection is configured on port 443 on the IIS server that hosts Datacap Web Server.

Datacap Web configuration

Datacap Web integration with WebSEAL requires the use of an LDAP bind. The Datacap LLDAP plugin that is used to retrieve the LDAP group information of the user requires the following LDAP information:

- LDAP Server ID and Port Number
- LDAP Bind User ID and Password
- Group Search Filter Name
- Group Base Domain Name (DN)
- User Search Filter Name
- User Base Domain Name (DN)

This LDAP information is used to populate the Datacap Server LLDAP Authentication path template. For example:

```
Server:389/BindUser:cn=binduser?BindPw:mypassword?ValidateUser:Off?
UserBaseDn:cn=mydomain?UserSearchFilter:
  (&(objectClass=organizationalPerson)(uid=<%user%>)) ?UserShortNameAttr:cn?
UserDisplayNameAttr:uid?GroupBaseDn:cn=mydomain
?GroupSearchFilter:(&(objectClass=groupOfNames)) ?GroupShortNameAttr:cn?
GroupDisplayNameAttr:cn
?GroupMembershipSearchFilter:(&(objectClass=groupOfNames)(member=<%user%>))
```

Attention: The ValidateUser:Off parameter directs the LLDAP plug-in to skip authentication of the users credentials and move to group retrieval. This flag is optional and is not required for WebSEAL integration. If this flag is enabled, it applies to all of the users who are authenticated through the Datacap Server that are using LLDAP.

Datacap Security Access Manager authentication process

Datacap users are authenticated by using the iv-user header value in Security Access Manager instead of the user name and password. The WebSEAL server controls the access to the Datacap Web URLs. After a successful authentication of the user credentials, WebSEAL forwards the Datacap Web URL to the user.

The following steps describe the authentication process:

1. Datacap Web aspx page extracts user identity information from the iv-user value in the HTTP header and passes the user id to Datacap Server. The Datacap Server processes the user id and passes it to Datacap LLDAP authentication plugin.
2. The Datacap LLDAP plugin retrieves the LDAP group information of the user by using the directory information that is listed in Datacap LLDAP authentication template.
3. Once the LDAP group information is retrieved, the Datacap LLDAP plugin populates the group list and passes it to the Datacap Server.
4. The Datacap Server validates the group list against the Administration database.
5. Datacap Web presents the user with the login page with the user id and password fields greyed out. Only the application and station fields are enabled. Once the user has selected the application, the application name is matched against the user groups in the Administration database and the user is presented with the validated workflows.

Switching between Datacap applications

To switch between Datacap applications:

1. Click Logout on the Datacap page. The Datacap Login page is displayed with the User ID and Password fields greyed out. Select another Datacap application and enter the Station Number. WebSEAL passes the same iv-user header value to the Datacap Web Login page where you are authenticated against the selected Datacap application.

Installing the Datacap modules

You must copy the new Datacap dll files over the existing Datacap installation.

1. Copy the new dctmlll.dll file with the LLDAP changes to the C:\Datacap directory.
2. Open a command window and run the following command:

```
regsvr32 dctmlll.dll
```

3. Copy the new App_Web_edlogin.ascx.cdca7d2.dll file with the Datacap Web changes to the C:\Datacap\tmlweb.net\bin directory.

Parent topic: [Configuring authentication for Datacap](#)

Configuring single sign-on (SSO) for Datacap Navigator

»You can configure single sign-on (SSO) for Datacap Navigator by using IBM Security Access Manager, SPNEGO/Kerberos, or container-managed authentication.«

»

About this task

Datacap Navigator supports the following authentication methods:

IBM Security Access Manager

You can configure SSO for Datacap Navigator with IBM Security Access Manager. First, you need to set up SSO in IBM® Content Navigator. For more information, see [Configuring single sign-on for IBM Content Navigator by using IBM Security Access Manager for Web on WebSphere Application Server \(FileNet P8\)](#).

SPNEGO/Kerberos

Datacap Navigator supports SPNEGO/Kerberos authentication. In addition to configuring IBM Content Navigator for SSO, you must configure your browser. For more information, see [Configuring single sign-on for IBM Content Navigator by using SPNEGO/Kerberos on WebSphere Application Server \(IBM FileNet P8\)](#).

Container-managed authentication

Using container-managed authentication, you can switch between repositories in the same instance of IBM Content Navigator without logging in each time you access a repository. The container authentication user is used to communicate with the Datacap application. For example, you can authenticate initially in either IBM Content Navigator or Datacap Navigator, and then switch between the two applications without logging in again.

For more information about configuring IBM Content Navigator for SSO, see [Support for Single Sign-on \(SSO\)](#)

Restriction: For Datacap Navigator, SSO is supported for IBM WebSphere® Application Server only.

«

Procedure

To configure SSO for a Datacap repository:

1. Set up LLDAP authentication for Datacap Server; for instructions, see [LLLDAP group authentication](#).
2. Enable SSO for your application.
 - a. Access the IBM Content Navigator administration tool. In a browser, enter a URL with the following format:

```
http://host_name:port_number/context_root/?desktop=admin
```

By default, the context root is `navigator`.

- b. Click Repositories in the left pane.
- c. Select an application in the right pane and click Edit.

- d. Click Connect and log in to the application.
- e. Click the Configuration Parameters tab.
- f. Click Enable for the Single sign-on parameter.
- g. Click Save and Close.

Parent topic: [Configuring authentication for Datacap](#)

Installing and configuring IBM Datacap Advanced Handwriting Recognition 9.0.1

The IBM® Datacap Advanced Handwriting Recognition 9.0.1 is an add-on for handwriting recognition for checks.

You can install and configure the IBM Datacap Advanced Handwriting recognition 9.0.1 and migrate the existing ICR_P actions from the earlier versions to Datacap 9.0.1.

With IBM Datacap Advanced Handwriting Recognition you can:

- Process a check with machine printed and handwritten data.
- Extract data for supported data sets for supported countries (Brazil, Canada, France, India, United Kingdom and United States).
- Detect and validate signatures on checks.
- Recognize cursive handwriting.
- [Installing IBM Datacap Advanced Handwriting Recognition 9.0.1](#)
IBM Datacap Advanced Handwriting Recognition 9.0.1 is a separately licensed component that provides intelligent character recognition (ICR) and Check Processing & Signature Validation to IBM Datacap 9.0.1.
- [Using IBM Datacap Advanced Handwriting Recognition 9.0.1](#)
You can use the IBM Datacap Advanced Handwriting Recognition features using the action libraries.
- [Migrating from ICR_P to Advanced Handwriting Recognition actions](#)
The versions prior to IBM Datacap 9.0.1 used the ICR_P actions to recognize the contents within zoned fields that are configured for recognition. These actions use the Parascript FieldScript for IBM Datacap recognition engine. The Parascript FieldScript can be installed with earlier versions of IBM Datacap but not with new version of IBM Datacap 9.0.1.

Parent topic: [Datacap installation and configuration in a client/server environment](#)

Installing IBM Datacap Advanced Handwriting Recognition 9.0.1

IBM® Datacap Advanced Handwriting Recognition 9.0.1 is a separately licensed component that provides intelligent character recognition (ICR) and Check Processing & Signature Validation to IBM Datacap 9.0.1.

Before you begin

You must install or upgrade to IBM Datacap 9.0.1.

About this task

You install IBM Datacap Advanced Handwriting Recognition 9.0.1 on all computers that process rules, such as the Rulerunner server.

Procedure

To download and install IBM Datacap Advanced Handwriting Recognition 9.0.1:

1. Sign in to [Passport Advantage](#) by using your IBM ID. If you are not a returning customer or you do not have an IBM ID, you can register to obtain an IBM ID.
2. Download the IBM Datacap Advanced Handwriting Recognition 9.0.1 installation program.
3. Start the installation program on a computer that process rules, such as Rulerunner:
4. Choose which options you want to install:
Important: For the check processing feature to work correctly, you can install only one country in addition to IBM Datacap Check Processing (US). Do not install more than one country in addition to the US option.
 - IBM Datacap Cursive Recognition
 - IBM Datacap Cursive Recognition for Non-Production Environment
 - IBM Datacap Check Processing (US)
 - IBM Datacap Check Processing (UK)
 - IBM Datacap Check Processing (Brazil)
 - IBM Datacap Check Processing (Canada)
 - IBM Datacap Check Processing (India)
 - IBM Datacap Check Processing (France)
 - IBM Datacap Signature Validation
 - IBM Datacap Signature Validation for Non-Production Environment
5. Complete the installation.

What to do next

Install the program with the same options on every computer that processes rules.

Parent topic: [Installing and configuring IBM Datacap Advanced Handwriting Recognition 9.0.1](#)

Using IBM Datacap Advanced Handwriting Recognition 9.0.1

You can use the IBM® Datacap Advanced Handwriting Recognition features using the action libraries.

Before you begin

You must install or upgrade to IBM Datacap 9.0.1.

What to do next

For more details on using the Advanced Handwriting Recognition actions, refer to the following actions:

- [CheckProcessing actions](#)
- [HandwritingRecognition actions](#)
- [SignatureValidation actions](#)

Parent topic: [Installing and configuring IBM Datacap Advanced Handwriting Recognition 9.0.1](#)

Migrating from ICR_P to Advanced Handwriting Recognition actions

The versions prior to IBM® Datacap 9.0.1 used the ICR_P actions to recognize the contents within zoned fields that are configured for recognition. These actions use the Parascript FieldScript for IBM Datacap recognition engine. The Parascript FieldScript can be installed with earlier versions of IBM Datacap but not with new version of IBM Datacap 9.0.1.

Before you begin

You must install or upgrade to IBM Datacap 9.0.1.

About this task

During migration, you need to modify the application with the new actions which match old actions and discard the actions which do not match.

Procedure

To migrate from ICR_P to Advanced Handwriting Recognition actions:

1. Install IBM Datacap Advanced Handwriting Recognition 9.0.1.
2. Open the application using Datacap Studio or Fastdoc.
3. Replace/Discard the actions of ICR_P library as per following table:

	IBM Datacap 9.0.0	IBM Datacap 9.0.1
Library	icr_p	HandwritingRecognition
Actions	AddWord	Not supported
	DeleteWord	Not supported
	ImportCSF	Not supported
	LoadFromFile	SetVocabulary
	NewDictionary	Not supported
	RecognizeFieldsICR_P	Recognize
	SaveToFile	Not supported
	SetPostalDBPathICR_P	SetPostalDatabase

4. Save the application.
5. Run the application to test functionality.

Note: The new version does not support some actions, which may lead to loss of functionality.

Parent topic: [Installing and configuring IBM Datacap Advanced Handwriting Recognition 9.0.1](#)

Datacap Web Services installation steps

The Datacap Web Services installation in a client/server environment requires you to follow several steps, including preparation, installation, configuration, and verification steps.

You can install Datacap Web Services on a dedicated web server or on a web server on which other Datacap Web Services components are installed. You must have administrator access to all computers on which the Datacap software is installed and to the server on which you install Datacap Web Services.

The following list is an overview of the steps that are required to install and configure Datacap Web Services in a Datacap client/server environment.

- Ensure that you meet the prerequisites that are detailed in *Installation and configuration prerequisites*.
- You can adjust the value of the `maxAllowedContentLength` and `maxRequestLength` settings in the `web.config` file. The maximum size message that can be uploaded depends on your web server, proxy server, and client.
- Determine which Datacap applications you want Datacap Web Services to interact with and the type of interactions you want Datacap Web Services to have with Datacap.
- Ensure that you can run all of the application tasks manually before you configure Datacap Web Services to run them.
- Based on your authentication system, set up the Datacap Web Services authentication credentials in the Application Manager for users and stations. Access to the Datacap Web Services HTTP IP address and methods is limited to authorized computers and users by a firewall or other network devices.
- Set up security permissions for Datacap Web Services on the Datacap shared folder on the Datacap Server.
- Set up security permissions for Datacap Web Services account on the Datacap folder that contains the application with which Datacap Web Services is to be interacting.
- Install and configure Datacap Web Services on the web server, including selecting the appropriate language for installation purposes and selecting Datacap Web Service from the list of components.
- You can host Datacap Web Services as a Windows Service or you can set up IIS to host Datacap Web Services. Validate the Datacap Web Services installation by opening the help page.
 - To host Datacap Web Services as a Windows Service, select the Datacap Windows Service option in the installation wizard.
 - To set up IIS to host Datacap Web Services, you must add the wTM website, configure the application pool settings, and enable the ISAPI extension for All Verbs.
- Import encryption keys to from the Datacap Server to the computer on which you are installing Datacap Web Services to secure passwords that are sent across the network.
- Set the location of the `datacap.xml` file.
- [Datacap Web Services authentication](#)
The Datacap Web Services authenticates with the Datacap Server by using a configured user in the Application Manager or by calling the `\Session\Logon` endpoint and providing the credentials. When you use the Application Manager configured user, all authentication methods are supported. When you use the `\Session\Logon` endpoint, the TMA, ADLDS, and LLDAP authentication methods are supported.
- [Setting up sharing permissions for Datacap Web Services](#)
You must set up the sharing permissions for the Datacap Web Services account on the shared Datacap folder.
- [Setting up shared folder security for Datacap Web Services](#)
You must set up security permissions on the shared folder with which the Datacap Web Services account is interacting.
- [Setting up application folders security for Datacap Web Services](#)
You must set up security permissions on the application folders with which the Datacap web services account is interacting.
- [Installing Datacap Web Services](#)
You can install Datacap Web Services on the web server by using the installation wizard.
- [Datacap Web Services hosting options](#)
You can host Datacap Web Services as a Windows service or by configuring Datacap Web Services on Microsoft Internet Information Services (IIS).
- [Disabling security for the transaction endpoints](#)
You can disable security for the transaction endpoints in the `/Transaction/` URI path.
- [Enabling SSL for Datacap Web Services](#)
You can enable Secure Sockets Layer (SSL) for Datacap Web Services by modifying settings in the Datacap Web Services configuration file.
- [Importing encryption keys to Datacap computers](#)
In a client/server configuration, you must import security encryption keys to the computer where you

are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.

Parent topic: [Installing and configuring in a client/server environment](#)

Related concepts:

[Installation and configuration prerequisites](#)

Datacap Web Services authentication

The Datacap Web Services authenticates with the Datacap Server by using a configured user in the Application Manager or by calling the `\Session\Logon` endpoint and providing the credentials. When you use the Application Manager configured user, all authentication methods are supported. When you use the `\Session\Logon` endpoint, the TMA, ADLDS, and LLDAP authentication methods are supported.

The Datacap Web Services configured user, password, and station information is retrieved from the Application Manager for authentication. Configure the Application Manager Custom values settings with the name and value pair for the user, password, and station authentication. Datacap Web Services uses settings in the `c:\Datacap\wTM\web.config` file to determine the names of the keys that are stored in the Application Manager from which the user name, password, and station information is retrieved. You must set up the `wTMUser`, `wTMPassword`, and `wTMStation` name and value pairs in the Application Manager that is based on your authentication method. The `web.config` file contains the following lines that identify the names of the keys.

```
<setting name="pathUser" serializeAs="String">
  <value>values/gen/wTMUser</value>
</setting>
<setting name="pathPassword" serializeAs="String">
  <value>values/adv/wTMPassword</value>
</setting>
<setting name="pathStation" serializeAs="String">
  <value>values/gen/wTMStation</value>
</setting>
```

When you are using the Application Manager configured user, Datacap Web Services authenticates with the Datacap Server when each endpoint is called. Each time an endpoint is called, Datacap Web Services sends a request to the Datacap Server to log in the user, then processes the endpoint action, and logs off the user.

When you are using the `\Session\Logon` endpoint, a user session is persisted for subsequent calls to reduce the number of calls to the Datacap Server. When the user is logged in, the session between Datacap Web Services and the Datacap Server continues throughout all subsequent endpoints until the `\Session\Logoff` endpoint is called or the session expires.

To avoid unauthorized user access when you use the `\Session\Logon` endpoint, ensure that a user is not configured in the Application Manager. If you are using ADSI or LDAP for your application, configure a separate instance of the Datacap Server for Datacap Web Services to use the TMA, ADLDS, or LLDAP authentication method.

- [Configuring Datacap Web Services authentication](#)
To set the Datacap Web Services credentials when your authentication system is TMA, you must add the Datacap user, password, and station in the Application Manager.
- [Configuring Datacap Web Services ADSI or LDAP authentication](#)
To set the Datacap Web Services credentials when your authentication system is ADSI or LDAP, you must add the name and value pair for the user, password, and station in the Application Manager.
- [Configuring Datacap Web Services ADLDS or LLDAP authentication](#)
You can set the Datacap Web Services credentials when your authentication system is ADLDS or LLDAP by adding the user, password, and station in the Application Manager. If you are using the

\Session\Logon endpoint to provide valid credentials and authenticate, you must not add the user, password, and station in the Application Manager.

Parent topic: [Datacap Web Services installation steps](#)

Related tasks:

[Configuring authentication for Datacap](#)

Related reference:

[Session/Logon](#)

[Session/Logoff](#)

Configuring Datacap Web Services authentication

To set the Datacap Web Services credentials when your authentication system is TMA, you must add the Datacap user, password, and station in the Application Manager.

About this task

When your authentication system is TMA, follow this procedure to set up the Datacap Web Services credentials for your application. In Application Manager, add the name and value pair for a Datacap user, password, and station.

Procedure

1. Add a name and value pair in General string values section of the Custom values tab for the Datacap user name.
 - o Value name: `wTMUser`
 - o Value: Set to the Datacap user name.
2. Add a name and value pair in the General string values section of the Custom values tab for the Datacap station name.
 - o Value name: `wTMStation`
 - o Value: Set to the Datacap station name.
3. Add a name and value pair in the Advanced values section of the Custom values tab for the Datacap user password.
 - o Value name: `wTMPassword`
 - o Value: Set to the Datacap user password.

Parent topic: [Datacap Web Services authentication](#)

Configuring Datacap Web Services ADSI or LDAP authentication

To set the Datacap Web Services credentials when your authentication system is ADSI or LDAP, you must add the name and value pair for the user, password, and station in the Application Manager.

About this task

When your authentication system is ADSI or LDAP, follow this procedure to set up the Datacap Web Services credentials for your application. In the Application Manager, add the name and value pair for a user, password, and station.

Procedure

1. Add a name and value pair in General string values section of the Custom values tab for the blank user name.
 - o Value name: `wTMUser`
 - o Value: Leave this field blank.
2. Add a name and value pair in the General string values section of the Custom values tab for the Datacap station name.
 - o Value name: `wTMStation`
 - o Value: Set to the Datacap station name.
3. Add a name and value pair in the Advanced values section of the Custom values tab for the blank user password.
 - o Value name: `wTMPassword`
 - o Value: Leave this field blank.

Parent topic: [Datacap Web Services authentication](#)

Configuring Datacap Web Services ADLDS or LLDAP authentication

You can set the Datacap Web Services credentials when your authentication system is ADLDS or LLDAP by adding the user, password, and station in the Application Manager. If you are using the `\Session\Logon` endpoint to provide valid credentials and authenticate, you must not add the user, password, and station in the Application Manager.

About this task

When your authentication system is ADLDS or LLDAP, you can configure the user in the Application Manager or you can use the `\Session\Logon` endpoint for authentication.

When you use the Datacap Web Services `\Session\Logon` method, you force each client of the web service to provide credentials and authenticate. The session between Datacap Web Services and the Datacap Server continues throughout all subsequent endpoints until the `\Session\Logoff` endpoint is called or the session expires. If you use the `\Session\Logon` method, you must not configure the user in the Application Manager.

If you configure the user, password, and station in the Application Manager, Datacap Web Services authenticates with the Datacap Server when an endpoint is called. Each time an endpoint is called, Datacap Web Services sends a request to the Datacap Server to log in the user, then processes the endpoint action, and logs off the user.

Follow this procedure to set up the Datacap Web Services credentials for your application in the Application Manager. Add the name and value pair for a user, password, and station. The user name and password must match the user name and password that is set up in your ADLDS or LLDAP authentication system.

Procedure

1. Add a name and value pair in General string values section of the Custom values tab for the Datacap Web Services user name.
 - o Value name: `wTMUser`
 - o Value: Enter the user name.
2. Add a name and value pair in the General string values section of the Custom values tab for the Datacap Web Services station name.
 - o Value name: `wTMStation`
 - o Value: Enter the station name.

3. Add a name and value pair in the Advanced values section of the Custom values tab for the Datacap Web Services user password.
 - o Value name: wTMPassword
 - o Value: Enter the password.

Parent topic: [Datacap Web Services authentication](#)

Setting up sharing permissions for Datacap Web Services

You must set up the sharing permissions for the Datacap Web Services account on the shared Datacap folder.

About this task

An example of the shared folder is the C:\Datacap folder on the Datacap Server. Note that other accounts were already granted sharing permissions during the initial installation and configuration of the Datacap Server.

Procedure

1. On the server, start Windows Explorer, navigate to, and right-click the c:\Datacap folder and select Properties.
2. Click the Sharing tab and ensure that the folder is shared with the Share name of Datacap.
3. Click Advanced Sharing. When User Account Control (UAC) is on, the User Account Control window is displayed. Click Yes.
4. Click Permissions. Ensure that the domain/Windows user ID of Datacap Web Services is set to allow Full Control.

Parent topic: [Datacap Web Services installation steps](#)

Setting up shared folder security for Datacap Web Services

You must set up security permissions on the shared folder with which the Datacap Web Services account is interacting.

About this task

You must set up the security permissions for the Datacap Web Services account on the shared folder, such as c:\Datacap. In the examples that are used in this material, the shared folder is the c:\Datacap folder on the Datacap Server. Note that other accounts were already granted security permissions during the initial installation and configuration of the Datacap Server.

Procedure

1. On the server, start Windows Explorer, navigate to, and right-click the c:\Datacap folder and select Properties.
2. Click the Security tab and click Edit. When User Account Control (UAC) is on, the User Account Control window is displayed. Click Yes.
3. Ensure that the domain Windows user ID of the Datacap Web Services account is set to Read.

Parent topic: [Datacap Web Services installation steps](#)

Setting up application folders security for Datacap Web Services

You must set up security permissions on the application folders with which the Datacap web services account is interacting.

About this task

If you are setting up Datacap Web Services to run tasks from more than one application, you must set up security permissions for each Datacap\Application folder. Note that other accounts were already granted security permissions during the initial installation and configuration of the Datacap server. In the examples that are used in this material, the application files are found on the Datacap server in the shared c:\Datacap folder.

Procedure

1. On the server, start Windows Explorer, navigate to, and right-click the c:\Datacap\Application folder and select Properties.
2. Click the Security tab and click Edit.
3. Ensure that the domain Windows user ID of the Datacap Web Services account is set to Read.

Parent topic: [Datacap Web Services installation steps](#)

Installing Datacap Web Services

You can install Datacap Web Services on the web server by using the installation wizard.

About this task

Follow this procedure to install the Datacap Web Services software component on the web server. You can also install the Datacap Windows Service, if you want to host Datacap Web Services as a Windows service.

Procedure

1. Make the installation package available on your network or insert the Datacap CD in the web server's CD/DVD drive. If the installation process does not start automatically or if the package is on the network, open Windows Explorer, navigate to and double-click the Setup.exe.
2. Select the appropriate language, then click OK. The language that you select determines the language that is displayed by the installation program during the installation process.
3. When additional, redistributed software is required, the installation program displays a list of the items to be installed. Click Install.
4. Click Next.
5. Accept the license agreement and click Next.
6. Select the Custom option and click Next.
7. Select the Datacap Web Service option from the list of components.
8. Select the Datacap Windows Service option from the list of components, if you want to host Datacap Web Services as a Windows service.
9. Click Next.
10. Click Install.
11. Click Finish.

Parent topic: [Datacap Web Services installation steps](#)

Datacap Web Services hosting options

You can host Datacap Web Services as a Windows service or by configuring Datacap Web Services on Microsoft Internet Information Services (IIS).

To host Datacap Web Services as a Windows service, select the custom option and choose the Datacap Windows Service option in the installation wizard.

To host Datacap Web Services on Microsoft Internet Information Services (IIS), add the wTM website and configure the application pool settings.

- [Ensuring the required IIS components are installed](#)
This topic identifies the set of Microsoft Internet Information Services (IIS) Web Server Role Services that are required by the various Taskmaster web server components (Taskmaster Web, RV2, Fingerprint Service, or Taskmaster Web Services).
- [Setting up the Windows service to host Datacap Web Services](#)
To host Datacap Web Services as a Windows service, install the Datacap Windows Service by using the installation wizard. You must start the Datacap Windows Service before you can validate the installation by opening the Datacap Web Services help page.
- [Setting up IIS to host Datacap Web Services](#)
Set up Datacap Web Services on Microsoft Internet Information Services (IIS) by adding the wTM website, configuring the application pool settings, and enabling the ISAPI extensions for All verbs.

Parent topic: [Datacap Web Services installation steps](#)

Ensuring the required IIS components are installed

This topic identifies the set of Microsoft Internet Information Services (IIS) Web Server Role Services that are required by the various Taskmaster web server components (Taskmaster Web, RV2, Fingerprint Service, or Taskmaster Web Services).

About this task

This procedure provides instructions on how to ensure that the appropriate components are installed in Microsoft Internet Information Services (IIS) when the WebServer's operating system is Microsoft Windows Server 2008 R2 or Microsoft Windows Server 2012.

Procedure

1. From the WebServer's Windows Start menu, select Administrative Tools > Server Manager.
2. In Server Manager, expand Roles, and then select Web Server (IIS).
3. In Web Server (IIS), expand Role Services, and under Common HTTP Features, ensure that Static Content, Default Document, Directory Browsing, and HTTP Errors are installed.
Important: If you are installing wTM, do not install the WebDAV Publishing role service, as it prevents the wTM PUT method from functioning.
4. In the Role Services pane, under Application Development, ensure that the ASP.NET, .NET Extensibility, ASP, ISAPI Extensions, and ISAPI Filters are installed.
5. In the Role Services pane, under Health and Diagnostics, ensure that HTTP Logging and Request Monitor are installed.
6. In the Role Services pane, under Security, ensure that Request Filtering is installed.
7. In the Role Services pane, under Performance, ensure that Static Content Compression is installed.
8. In the Role Services pane, under Management Tools, ensure that IIS Management Console is installed.
9. Close the Server Manager window.

Setting up the Windows service to host Datacap Web Services

To host Datacap Web Services as a Windows service, install the Datacap Windows Service by using the installation wizard. You must start the Datacap Windows Service before you can validate the installation by opening the Datacap Web Services help page.

Before you begin

Install Datacap Web Services by using the installation wizard.

About this task

Follow this procedure to install and start the Datacap Windows Service to host the Datacap Web Services.

Procedure

1. Install the Datacap Windows Service from the installation wizard.
 - a. Make the installation package available on your network or insert the Datacap CD in the web server's CD/DVD drive. If the installation process does not start automatically or if the package is on the network, open Windows Explorer, navigate to and double-click the Setup.exe.
 - b. Select the appropriate language, then click OK. The language that you select determines the language that is displayed by the installation program during the installation process.
 - c. When additional, redistributed software is required, the installation program displays a list of the items to be installed. Click Install.
 - d. Click Next.
 - e. Accept the license agreement and click Next.
 - f. Select the Custom option and click Next.
 - g. Select the Datacap Windows Service option from the list of components.
 - h. Click Next.
 - i. Click Install.
 - j. Click Finish
2. Start the Datacap Windows Service.
 - a. Go to Administrative Tools > Services.
 - b. Right-click Datacap Windows Service and select Start.
3. Validate the installation of Datacap Web Services by opening the help page.
 - a. Go to c:\Datacap\Taskmaster and open wTMservice.exe.config with Notepad.
 - b. Copy the `baseAddress` URL to your browser and add `/help`. The following is an example of a `baseAddress` URL in the wTMservice.exe.config file.

```
<host>
  <baseAddresses>
    <add baseAddress="http://localhost:82/service" />
  </baseAddresses>
</host>
```

- In this example, the URL of the help page is `http://localhost:82/service/help`,
- c. When the Datacap Web Services help page opens, you can click one of the links in the Method column to show detailed information about the REST API endpoint.

Setting up IIS to host Datacap Web Services

Set up Datacap Web Services on Microsoft Internet Information Services (IIS) by adding the wTM website, configuring the application pool settings, and enabling the ISAPI extensions for All verbs.

About this task

Follow this procedure to set up the Datacap Web Services wTM website on Microsoft Internet Information Services (IIS).

Procedure

1. Add the Datacap Web Services in IIS.
 - a. From Start, select Administrative Tools > Internet Information Services (IIS) Manager.
 - b. In the Connections pane, right-click Sites and select Add Web Site.
 - c. Set Site name to wTM. The Application pool is automatically set to wTM.
 - d. Set the Physical path by entering or browsing to the installation folder for Datacap Web Services. The default location is C:\Datacap\wTM.
 - e. Select the IP address of the Datacap Web Services server and assign a unique Port number and click OK to close the Add Web Site dialog.

Note: The Datacap Web Client and Datacap Web Services must be assigned different port numbers when they are installed on the same server. The Datacap Web Client does not work unless Datacap Web Services is assigned a different port number.
2. Set up the Datacap Web Services IIS application pool.
 - a. In the Connections pane, select Application Pools.
 - b. In the Application Pools pane, select the wTM application pool. Select Actions > Edit Application Pool > Advanced Settings.
 - c. Ensure that the .NET Framework Version is set to v4.0.
 - d. Ensure that Enable 32-Bit Applications is set to True.
 - e. Ensure that Managed Pipeline Mode is set to Integrated.
 - f. Ensure that Start Automatically is set to True.
 - g. In the Process Model section, click the Browse button next to Identity.
 - h. In the Application Pool Identity window, select Custom account and click Set.
 - i. In the Set Credentials window, enter the wTM domain Windows account information in the format, accountname@domainname. Enter the account password and click OK.
 - j. In the Process Model section, set Load User Profile to True and click OK.
 - k. In the Connections pane, select the wTM site. In the Actions pane under Manage Web Site, click Restart.
 - l. Confirm that the Application Pools, Default Web Site, and wTM website are started.
3. Enable ISAPI extensions for All verbs and for execution on the Datacap Web Services server.
 - a. In the Connections pane, select the wTM site. In the wTM Home pane, double-click Handler Mappings.
 - b. Scroll down, select svc-ISAPI-4.0_32bit, and ensure that it is enabled.

Tip: For Windows Server 2012 R2, the svc-ISAPI-4.0_32bit setting might be missing. To resolve this problem, in your Windows system administrative tools, open Server Manager and click Dashboard. Click Add roles and features. In the Add Roles and Features Wizard, go to the Features section. In .NET Framework 4.5 Features > WCF Services, select the HTTP Activation check box.
 - c. In the Actions pane, click Edit Feature Permissions, select Read, Script, and Execute, then click OK.
 - d. In the Actions pane, click Edit, and on the Edit Script Map dialog, click Request Restrictions.
 - e. On the Request Restrictions dialog, click the Verbs tab, select All verbs, click OK, then click Yes.

4. Validate the Datacap Web Services installation by opening the Datacap Web Services help page.
 - a. Open Internet Explorer and enter the following help page URL, `http://<WebServerName or IP address>:<port number>/ServicewTM.svc/help`.
 - b. Click a link in the Method column to show detailed information about the REST API endpoint.

Parent topic: [Datacap Web Services hosting options](#)

Related concepts:

[Installation and configuration prerequisites](#)

Disabling security for the transaction endpoints

You can disable security for the transaction endpoints in the `/Transaction/` URI path.

About this task

Security for transactional endpoints is enabled by default. The `transactionSecurity` parameter is set to true by default. When true, you must first authenticate by using the `Session/Logon` endpoint.

When this parameter is set to false, the web service does not connect to Datacap Server for that endpoint.

Procedure

To disable security for the transaction endpoints:

1. Open the configuration file:
 - o When hosting the web services by using Microsoft Internet Information Services (IIS), open: `\Datacap\wTM\web.config`
 - o When hosting the web services as a Windows Service: `\Datacap\wTM\wTMservice.exe.config`
2. Set the value for `transactionSecurity` to false, as follows:

```
<setting name="transactionSecurity" serializeAs="String">  
<value>False</value>  
</setting>
```

Parent topic: [Datacap Web Services installation steps](#)

Enabling SSL for Datacap Web Services

You can enable Secure Sockets Layer (SSL) for Datacap Web Services by modifying settings in the Datacap Web Services configuration file.

Procedure

1. Import the SSL certificate.

For example, use the Microsoft Management Console (MMC) to import the certificate. After the import, verify that the certificate is trusted, that the private key is in the store, and that it is not expired.

2. Update the port. The port is set in the binding in Internet Information Services (IIS) or in the configuration file for the service.
 - o When hosting the web services by using IIS, see the following steps for configuring an IIS-hosted Windows Communication Foundation (WCF) service with SSL <https://msdn.microsoft.com/en-us/library/hh556232>

- o When hosting the web services as a Windows Service, see the following steps for configuring a port with an SSL certificate: <https://msdn.microsoft.com/en-us/library/ms733791>
3. Enable SSL for Datacap Web Services:
- o When hosting the web services by using Microsoft Internet Information Services (IIS), do the following steps:
 - a. Open \Datacap\wTM\web.config.
 - b. Add the SecureWebHttpBinding binding by changing the following line:

```
<webHttpBinding />
```

to:

```
<webHttpBinding>
  <binding name="SecureWebHttpBinding">
    <security mode="Transport">
      <transport clientCredentialType="Basic" />
    </security>
  </binding>
</webHttpBinding>
```

- c. Change the following line:

```
<serviceMetadata httpGetEnabled="true" />
```

to:

```
<serviceMetadata httpGetEnabled="false" httpsGetEnabled="true"/>
```

- d. Save your changes.

- o When hosting the web services as a Windows Service, do the following steps:
 - a. Open \Datacap\wTM\wTMservice.exe.config.
 - b. Change the following line:

```
<serviceMetadata httpGetEnabled="true" />
```

to:

```
<serviceMetadata httpGetEnabled="false" httpsGetEnabled="true"/>
```

- c. Change the following line:

```
<binding name="NewBinding1" maxBufferSize="104857600"
  maxReceivedMessageSize="104857600" />
```

to:

```
<binding name="NewBinding1" maxBufferSize="104857600"
  maxReceivedMessageSize="104857600">
  <security mode="Transport" />
</binding>
```

- d. Change http to https in the following attribute, as follows:

```
<add baseAddress="https://localhost:port/service" />
```

- e. Save your changes.

4. If you have a Datacap Navigator configuration, deploy a client certificate on IBM® WebSphere® Application Server for IBM Content Navigator:
- a. Log in to the WebSphere Application Server administrative console on the IBM Content Navigator server that acts as the SSL client.
 - b. Navigate to Security > SSL certificate and key management.

- c. In the Related Items section, click Key stores and certificates.
- d. Select the default truststore:

Table 1. Truststores for IBM WebSphere Application Server

Configuration Type	Truststore
IBM WebSphere Application Server Network Deployment	CellDefaultTrustStore
WebSphere Application Server base and standalone	NodeDefaultTrustStore

- e. In the Additional Properties section, click Signer certificates.
- f. Click Retrieve From Port.
- g. In the Host field, enter the hostname of the wTM server.
- h. In the Port field, enter the secure wTM port.
- i. In the Alias field, enter a name for this certificate.
- j. Click Retrieve Signer Information.
- k. Verify that the certificate information is for a certificate that you can trust.
- l. Click Apply, and click Save.
- m. Restart the IBM Content Navigator application.
- n. On IBM Content Navigator, update the Datacap repository with the Datacap Web Services HTTPS URL.
- o. On IBM Content Navigator, update the plug-in with the Datacap Web Services HTTPS URL.

Parent topic: [Datacap Web Services installation steps](#)

Importing encryption keys to Datacap computers

In a client/server configuration, you must import security encryption keys to the computer where you are installing and configuring each Datacap component. This requirement secures passwords that are sent across the network between Datacap servers and clients.

Before you begin

You must generate the encryption keys in the keystore on a server on which the Datacap server software component is installed. You export the new keys to a key transport file.

Procedure

To import encryption keys to Datacap computers:

1. Find the dc_KTF.xml key transport file in the c:\Datacap\Taskmaster folder on the Datacap server where you generated and exported the encryption keys.
2. Copy the dc_KTF.xml key transport file to the appropriate folder on the computer where you installed the new component. The encryption keys are automatically applied to the keystore the next time you start the Datacap component.

Table 1. Encryption key folder locations by component

Component	Folder

Component	Folder
Datacap Server	C:\Datacap\Taskmaster
Datacap applications	C:\Datacap\Taskmaster
Datacap Desktop	C:\Datacap\DcDesktop
Datacap Studio	C:\Datacap\DStudio
Datacap FastDoc	C:\Datacap\FastDoc
Datacap Report Viewer	C:\Datacap\RV2\bin
Datacap Web Client	C:\Datacap\tmweb.net\bin
Datacap Web Services	C:\Datacap\wTM\bin

Parent topic: [Datacap Web Services installation steps](#)

Client/server installation checklist

This checklist provides a summary of the account and configuration settings that are required to run Datacap software components in a client/server environment for demonstration, proof of concept, development and test purposes.

This configuration uses a number of Windows accounts that must be in the authentication provider directory, such as in a Window Active Directory domain. The implementer must have an account that has administrator rights on every computer on which Datacap is to be installed.

The sections are sequenced deliberately because some items in later sections depend on steps in the previous sections being completed successfully.

- [Datacap server setup](#)
The checklists to set up the Datacap server include the tasks that are required to install and configure Datacap on the server.
- [Configuring Datacap Web Server on a supported version of Windows Server](#)
The checklist to configure Datacap Web Server includes ensuring that the .NET Framework is installed, the application pool identity is set, and the encryption keys are imported.
- [Datacap developer workstation setup](#)
The checklist to set up the Datacap developer workstation include the tasks to ensure the .NET Framework and Datacap client components are installed.
- [Completing Datacap server setup](#)
The checklists to complete the Datacap server setup include adding the list of applications to the Datacap.xml file and setting security permissions on the application folder.
- [Complete the Datacap Web Client setup](#)
The checklist to complete the Datacap Web Client setup includes adding the location of the Datacap.xml file on the Datacap server.
- [Running a Datacap client on Developer Workstation](#)
You can quickly test and confirm that each job a workflow is operating correctly by starting the Datacap Server Service and running a task in Datacap Desktop
- [Remote workstation setup](#)
The checklist includes tasks to add the TMWeb.net address as a trusted site, configure and test Internet Explorer manually, and run the Datacap Web Client application. You can use the Web Client Configuration tool to create a package for a user to configure and test Internet Explorer at the remote workstation.
- [User workstation and permissions setup](#)
The checklist to set up the user workstation includes setting up sharing and security permissions,

importing encryption keys, and installing the Datacap client components.

- [Datacap Report Viewer setup](#)
To install Report Viewer in a client server environment, you must perform the required set up tasks on the server and the workstation.
- [Rulerunner installation and configuration](#)
The checklist includes tasks to set security and sharing permissions on the Datacap\RRS folder, install Rulerunner, and import encryption keys.
- [Fingerprint Service setup](#)
The checklist includes tasks to configure the Fingerprint service application pool and set up security on the Datacap and application folders.
- [Datacap Maintenance Manager setup](#)
You must complete the required account and configuration settings before you can run Datacap Maintenance Manager on Datacap in a client/server environment. You can run Maintenance Manager for demonstration, proof of concept, development, and test purposes.

Parent topic: [Installing and configuring in a client/server environment](#)

Datacap server setup

The checklists to set up the Datacap server include the tasks that are required to install and configure Datacap on the server.

- [Install the Datacap server](#)
The checklist includes setting up the installer's user ID, ensuring the .NET Framework is installed, and starting the installation wizard.
- [Configuring Datacap on the server](#)
The checklist includes configuring the Datacap server service, setting up security and sharing permissions on the Datacap folder, and exporting encryption keys.

Parent topic: [Client/server installation checklist](#)

Install the Datacap server

The checklist includes setting up the installer's user ID, ensuring the .NET Framework is installed, and starting the installation wizard.

On	Do This
Datacap server	<p>Set up the installer's user ID.</p> <p>Add or ensure that the domain/Windows user ID of implementer is in the Administrators group on every machine to allow the installer to run the installation program.</p>
Datacap server	<p>Ensure that Microsoft .NET Framework 3.5.1 is installed.</p> <ul style="list-style-type: none">• Start > Administrative Tools > Server Manager .• In the Server Manager hierarchy pane, select Features.• Click Add Features.• Select .NET Framework 3.5.1 Features, click Next.• Click Install.

On	Do This
Datacap server	<p>Start the installation program wizard by running setup.exe.</p> <p>Install the Datacap server component and separately licensed connectors to which you are entitled.</p>

Parent topic: [Datacap server setup](#)

Configuring Datacap on the server

The checklist includes configuring the Datacap server service, setting up security and sharing permissions on the Datacap folder, and exporting encryption keys.

About this task

On	Do This
Datacap server	<p>Datacap server service.</p> <p>Ensure that domain/Windows account exists for Datacap server service.</p> <p>Grant Datacap server service the Log On as a Service privilege:</p> <ol style="list-style-type: none"> 1. Go to Control Panel > Administrative Tools > Services and right-click Datacap Server. Select Properties and click the Log On tab of the Properties window. 2. Select This Account and enter the domain name, user name, and password of the Datacap server service user that you just set up. Click Apply to confirm that the user is granted the Log On as a Service privilege. <p>Start Datacap server service automatically on reboot:</p> <ol style="list-style-type: none"> 1. Go to Control Panel > Administrative Tools > Services and right-click Datacap Server. Select Properties and click the General tab of the Properties window. 2. Set the Startup Type to Automatic.

On	Do This
Datacap server	<p>Share Datacap folder and set up sharing permissions.</p> <ul style="list-style-type: none"> • Right-click C:\Datacap folder and select Properties. • Click Sharing tab, click Advanced Sharing. • Click Share this Folder (as Datacap). • Click Permissions button. • Add NETWORK SERVICE and local IUSR and allow Full Control. • Add or ensure that the domain/Windows user IDs of developers are set to allow Full Control. • Add or ensure that the domain/Windows user ID of Datacap server service is set to allow Full Control. • Add or ensure that the domain/Windows user ID of Datacap Web Client is set to allow Read. • When Batches folders are staying on the Datacap server in C:\Datacap\Application path, add or ensure that the domain/Windows user IDs of Datacap users are set to allow Full Control.
Datacap server	<p>Set up security on shared folder.</p> <ul style="list-style-type: none"> • Right-click C:\Datacap folder and select Properties. • Click Security tab. • Add or ensure that the domain/Windows user IDs of developers who can change existing Datacap applications are set to allow Read & Execute. This developer cannot create new applications. • Add or ensure that the domain/Windows user IDs of developers who can create new Datacap applications in the C:\Datacap folder are set to allow Full Control. • Add or ensure that the domain/Windows user ID of Datacap server service is set to allow Read & Execute. • Add or ensure that the domain/Windows user ID of Datacap Web Client is set to allow Read & Execute. • When Batches folders are staying on the Datacap server in C:\Datacap\Application path, add or ensure that the domain/Windows user IDs of Datacap users are set to allow Read & Execute.
Datacap server	<p>Set up security on Datacap\RRS.</p> <ul style="list-style-type: none"> • Right-click C:\Datacap\RRS folder and select Properties. • Click Security tab. • Add NETWORK SERVICE and local IUSR and allow Read & Execute.
Datacap server	<p>Generate and export encryption keys.</p> <ul style="list-style-type: none"> • Open the command prompt and navigate to C:\Datacap\Taskmaster. • Enter the following command: <code>dcskey.exe e <Enter></code>.

Parent topic: [Datacap server setup](#)

Configuring Datacap Web Server on a supported version of Windows Server

The checklist to configure Datacap Web Server includes ensuring that the .NET Framework is installed, the application pool identity is set, and the encryption keys are imported.

On	Do This
Datacap Server	<p>Ensure that a Domain/Windows Account exists for Datacap Web Server.</p> <p>When Datacap Web Server or Report Viewer are installed on the same web server, they must use the same Domain/Windows account.</p>
Datacap Web Server	<p>Ensure that Microsoft .NET Framework 3.5.1 is installed.</p> <p>Start > Administrative Tools > Server Manager</p> <p>In the Server Manager hierarchy pane, select Features</p> <p>Click Add Features.</p> <p>Select .NET Framework 3.5.1 Feature, click Next.</p> <p>Click Install</p>
Datacap Server	<p>Stop Datacap Server Service:</p> <p>On Windows 2008 IIS 7.5 Server: Click Start > All Programs > IBM® Datacap Services > Datacap Server Manager. In Datacap Server Manager, click Stop.</p> <p>On Windows Server 2012 R2: In your programs, click IBM Datacap Services > Datacap Server Manager. In Datacap Server Manager, click Stop.</p>
Datacap Web Server	<p>Ensure that the required IIS components are installed.</p> <p>Start > Administrative Tools > Server Manager</p> <p>Expand Roles, select Web Server (IIS)</p> <p>In the Web Server (IIS) pane, ensure that you have the following components:</p> <ul style="list-style-type: none"> • Under Common HTTP Features, ensure that Static Content is installed. • Scroll down to Role Services, under Application Development, ensure that the ASP.NET module is installed. • Under Application Development, ensure that the ASP module is installed. • Scroll down further to Management Tools, ensure that IIS Management Console is installed.
Datacap Web Server	<p>Install Datacap Web Server on the web server.</p> <p>Install Datacap Web Server component and licensed connectors.</p>

On	Do This
Datacap Web Server	<p>Import encryption keys</p> <p>Navigate to the Datacap server on which you generated keys</p> <p>Copy the dc_KTF.xml file from Datacap server to the C:\Datacap\Taskmaster folder on the web server.</p>
Datacap Web Server	<p>Create the Datacap website.</p> <p>In the Start menu, click IBM Datacap Web > Datacap Web Client Configuration Tool.</p> <p>Ensure that the setting values are appropriate or change them.</p> <ul style="list-style-type: none"> • Select alternate site if you set one up. • Adjust ASP.NET Maximum File Upload Size and ASP Maximum Requesting Entity Body Limit. • Change App Pool Recycling Schedule. • Change Connection time-out. <p>Click Configure, click OK, click Exit.</p> <p>Start > Administrative Tools > IIS Manager.</p> <p>Expand Sites > Default Web Site in Connections pane, select tmweb.net site.</p> <p>Double-click Session State. In the Cookie Settings pane, enter tmweb in the Name field.</p> <p>In the Connections pane, select the Default Web Site. In the Actions pane under Manage Web Site, click Restart.</p>
Datacap Web Server	<p>Set Datacap Web Server Application Pool Identity.</p> <p>Start > Administrative Tools > IIS Manager.</p> <p>Select Application Pools. In the Actions pane, click Advanced Settings.</p> <p>In Process Model section, next to Identity, click browse, select Custom account, click Set</p> <p>Enter the Datacap Web Server domain/Windows account, such as accountname@domainname and password.</p> <p>Ensure that Enable 32-Bit Applications is set to True.</p> <p>In Process Model section, set Load User Profile to True.</p> <p>Click OK</p> <p>Ensure that Web Server, Application Pool, and Default Web Site are all started.</p>

On	Do This
Datacap Server	Start Datacap Server Service: Click IBM Datacap Services > Datacap Server Manager. In Datacap Server Manager, click Start.
Datacap Web Server	Change the Secure Sockets Layer (SSL) Setting in Server.INI. If using website SSL, also do the following: Using Notepad, open c:\Datacap\tmweb.net\server.ini and ensure UseSSL=1.

Parent topic: [Client/server installation checklist](#)

Datacap developer workstation setup

The checklist to set up the Datacap developer workstation include the tasks to ensure the .NET Framework and Datacap client components are installed.

On	Do This
Developer workstation	Ensure that Microsoft .NET Framework 3.5.1 is installed. Start > Control Panel > Programs > Programs and Features. Select Turn Windows features on or off. Select Microsoft .NET Framework 3.5.1 and click OK. Reboot the machine, if prompted.
Developer workstation	Run installation program wizard. Install Datacap client components, including the separately licensed applications and connectors to which you are entitled. Ensure FastDoc, Datacap Studio, and Maintenance Manager components are included.
Developer workstation	Import encryption keys. Navigate to the Datacap Server on which you generated keys. Copy the dc_KTF.xml file from Datacap server to the C:\Datacap\Taskmaster folder on the developer workstation.

- [Add tmweb.net address as trusted site](#)
The checklist to set up the Datacap server includes the task to add the Datacap Web Client tmweb.net address as a trusted site.
- [Configuring and testing IE manually](#)
The checklist includes the tasks to configure and test Internet Explorer manually, if you do not have access to the Datacap Web Client Configuration tool.
- [Copy the application to the Datacap server](#)
The checklist to copy an application to the Datacap server includes tasks for accessing the application wizard in Datacap Studio and setting the location of the Datacap.xml file.

Parent topic: [Client/server installation checklist](#)

Add tmweb.net address as trusted site

The checklist to set up the Datacap server includes the task to add the Datacap Web Client tmweb.net address as a trusted site.

On	Do This
Developer workstation, Workstation	<p>Add tmweb.net address as trusted site.</p> <p>Start IE > Tools > Internet Options > Security tab > Select Trusted sites > Click Sites button > Trusted sites dialog.</p> <p>On Trusted sites dialog, if the Datacap Web Client URL starts with http:, clear Require server verification, and add either the default IP address of the Datacap Web Client server (http://localhost) or the web server IP address or name as it is defined in your network DNS (http://WebServerName or https://WebServerName). Click Add.</p>

Parent topic: [Datacap developer workstation setup](#)

Configuring and testing IE manually

The checklist includes the tasks to configure and test Internet Explorer manually, if you do not have access to the Datacap Web Client Configuration tool.

About this task

If you do not have access to the Datacap Web Client Configuration tool, you can configure Internet Explorer manually.

On	Do This
----	---------

On	Do This
Developer workstation, Workstation	<p>Configure IE.</p> <p>Start Internet Explorer.</p> <p>Tools > Internet Options > Security tab.</p> <p>Select Trusted sites > click Sites button.</p> <p>On Trusted sites dialog, clear Require server verification, and add either the default IP address of the Datacap web server (http://127.0.0.1) or the URL of the web server as it is defined in your network's DNS (http://WebServerName or https://WebServerName).</p> <p>Security tab > Click Custom level button > Security settings dialog - Trusted Sites Zone:</p> <p>Enable: Download signed ActiveX controls and Initialize and script ActiveX controls not marked as safe for scripting.</p> <p>Scroll down to Miscellaneous settings > Enable Include local directory path when uploading files to a server.</p> <p>Click OK, then close IE.</p>
Developer workstation, Workstation	<p>Test IE.</p> <p>Start IE, enter the appropriate web server URL and test page (http://127.0.0.7/tmweb.net/ietest.aspx, http://WebServerName/tmweb.net/ietest.aspx, or https://WebServerName/tmweb.net/ietest.aspx).</p> <p>Click Yes to download Datacap TIFF Viewer, Thumbnails, and DataEdit Controls.</p> <p>Click Test. The red Xs change to green check marks.</p>

Parent topic: [Datacap developer workstation setup](#)

Copy the application to the Datacap server

The checklist to copy an application to the Datacap server includes tasks for accessing the application wizard in Datacap Studio and setting the location of the Datacap.xml file.

On	Do This
----	---------

On	Do This
Datacap Web Client server	Sharing the Datacap and tmweb.net folders on the Datacap Web Client server. <ul style="list-style-type: none"> • On the Datacap Web Client server, Datacap > Properties > Sharing > Advanced Sharing. • Click Share this Folder, keep <code>Datacap</code> as Share Name, click Permissions. • Add or ensure that the domain/Windows user ID of the developer is set to Full Control. • On the Datacap Web Client server, Datacap\TMWeb.Net > Properties > Security > Edit. • Add or ensure that the domain/Windows user ID of developer is set to Full Control.
Developer workstation	Start Datacap Studio to access the Datacap application wizard. <ul style="list-style-type: none"> • In the Start menu, click IBM Datacap Developer ToolsDatacap Studio > Close.
Developer workstation	Ensure that the correct Datacap.xml file is in use. <ul style="list-style-type: none"> • In Datacap Studio, click Settings. • Click the DCapp tab. • Set the path to the local datacap.xml. • Close and restart Datacap Studio.
Developer workstation	Copy the TravelDocs application to the Datacap server. <ul style="list-style-type: none"> • Click Datacap application wizard > Next > Copy an existing RRS application > Next. • Application: <code>TravelDocs</code>. • Root folder on target system: <code>\\Server\Datacap</code>. • Datacap Web Client folder on the web server: <code>\\WebServer\Datacap\TMWeb.Net</code>. • Rename Copy: Not applicable. • New Name: Not applicable. • Click Next and Finish.
Developer workstation	Click View Logs to open log in <code>\\Server\Datacap\TravelDocs</code> and search for errors and warnings that you need to fix.
Developer workstation	Copy datacap.xml from the developer workstation to the Datacap server. <ul style="list-style-type: none"> • Windows Explorer > <code>C:\Datacap</code>. • Windows Explorer > <code>\\Server\Datacap</code>. • Copy datacap.xml file from <code>C:\Datacap</code> to <code>\\Server\Datacap</code>.

On	Do This
Developer workstation	Set the location of the Datacap server and the datacap.xml file. <ul style="list-style-type: none"> • In the Start menu, click IBM Datacap Services> Datacap Application Manager • Select TravelDocs. Ensure that the paths are correct on the Main tab and set Locale. • Click the Datacap tab. Change the server name and address to the correct Datacap server and set Protocol. • Click Service tab. Ensure that the path is the correct location of the datacap.xml file on the Datacap server.

Parent topic: [Datacap developer workstation setup](#)

Completing Datacap server setup

The checklists to complete the Datacap server setup include adding the list of applications to the Datacap.xml file and setting security permissions on the application folder.

- [Update Datacap.xml on the server](#)
The checklist includes adding the list of applications and their locations to the Datacap.xml file.
- [Set application folder security permissions](#)
The checklist to add security to the application folder includes setting user permissions.

Parent topic: [Client/server installation checklist](#)

Update Datacap.xml on the server

The checklist includes adding the list of applications and their locations to the Datacap.xml file.

On	Do This
Datacap server	Update the Datacap.xml file. <ul style="list-style-type: none"> • Make a list of the applications that exist and their locations on the server. • Open \\Server\Datacap\datacap.xml in Notepad. • Edit the contents of datacap.xml to list only those applications that exist and are configured for use. The contents are case-sensitive.

Parent topic: [Completing Datacap server setup](#)

Set application folder security permissions

The checklist to add security to the application folder includes setting user permissions.

On	Do This
----	---------

On	Do This
Datacap server	<p>Set up security on Datacap\Application folder.</p> <p>Right-click c:\Datacap\Application folder and select Properties.</p> <p>On the Security tab, add NETWORK SERVICE and local IUSR and set to allow Full Control.</p> <p>Add or ensure that the domain/Windows user IDs of developers are set to allow Full Control.</p> <p>Add or ensure that the domain/Windows user ID of Datacap Web Client is set to allow Read & Execute.</p> <p>Add or ensure that the domain/Windows user ID of Datacap Server Service is set to allow Full Control.</p> <p>When Batches folders are staying on the server in C:\Datacap\Application path, add or ensure that the domain/Windows user IDs of Datacap users are set to allow Full Control.</p>

Parent topic: [Completing Datacap server setup](#)

Complete the Datacap Web Client setup

The checklist to complete the Datacap Web Client setup includes adding the location of the Datacap.xml file on the Datacap server.

On	Do This
Datacap Web Client server	<p>Set the location of the Datacap.xml file.</p> <ul style="list-style-type: none"> In the Start menu, click IBM Datacap Services>Datacap Application Manager Click Service tab. Ensure the path is the correct location of the datacap.xml on the Datacap server.
Datacap Web Client server	Restart Internet Information Services (IIS)

Parent topic: [Client/server installation checklist](#)

Running a Datacap client on Developer Workstation

You can quickly test and confirm that each job a workflow is operating correctly by starting the Datacap Server Service and running a task in Datacap Desktop

On	Do This
Server	<p>Start Datacap Server Service</p> <p>In the Start menu, click IBM Datacap Services> Datacap Server Manager> Start</p>

On	Do This
Developer Workstation	Test client Application Start > Programs > IBM Datacap Clients > Datacap Desktop Run and test the client and confirm that each job in the workflow is processing correctly Exit client

Parent topic: [Client/server installation checklist](#)

Remote workstation setup

The checklist includes tasks to add the TMWeb.net address as a trusted site, configure and test Internet Explorer manually, and run the Datacap Web Client application. You can use the Web Client Configuration tool to create a package for a user to configure and test Internet Explorer at the remote workstation.

- [Packaging the Web Client Configuration tool](#)
The checklist includes the tasks to create a package for a user to configure Internet Explorer on a remote workstation to access Datacap Web Client.
- [Add tmweb.net address as trusted site](#)
The checklist to set up the Datacap server includes the task to add the Datacap Web Client tmweb.net address as a trusted site.
- [Configuring and testing IE using a package](#)
The checklist includes the tasks to create a package for configuring and testing a remote connection with Internet Explorer.
- [Configuring and testing IE manually](#)
The checklist includes the tasks to configure and test Internet Explorer manually, if you do not have access to the Datacap Web Client Configuration tool.
- [Run the Datacap Web Client application](#)
The checklist to run the Datacap Web Client application includes starting the Datacap server service and entering the appropriate web server URL address on your browser.

Parent topic: [Client/server installation checklist](#)

Packaging the Web Client Configuration tool

The checklist includes the tasks to create a package for a user to configure Internet Explorer on a remote workstation to access Datacap Web Client.

About this task

On	Do This
----	---------

On	Do This
Any machine with Datacap installed	<p>Package the Web Client Configuration tool.</p> <p>Open C:\Datacap\support\WebConfiguration folder.</p> <p>Make backup copy of WebClientConfig.exe.config file.</p> <p>Open WebClientConfig.exe.config file and change server name in <value>http://localhost/tmweb.net</value>.</p> <p>Save and close .config file.</p> <p>Send WebClientConfig.exe.config, Datacap.Config.dll, and WebClientConfig.exe to the user with instructions to add the TMWeb.net address as trusted site, and configure and test Internet Explorer using the package.</p>

Parent topic: [Remote workstation setup](#)

Add tmweb.net address as trusted site

The checklist to set up the Datacap server includes the task to add the Datacap Web Client tmweb.net address as a trusted site.

On	Do This
Developer workstation, Workstation	<p>Add tmweb.net address as trusted site.</p> <p>Start IE > Tools > Internet Options > Security tab > Select Trusted sites > Click Sites button > Trusted sites dialog.</p> <p>On Trusted sites dialog, if the Datacap Web Client URL starts with http:, clear Require server verification, and add either the default IP address of the Datacap Web Client server (http://localhost) or the web server IP address or name as it is defined in your network DNS (http://WebServerName or https://WebServerName). Click Add.</p>

Parent topic: [Remote workstation setup](#)

Configuring and testing IE using a package

The checklist includes the tasks to create a package for configuring and testing a remote connection with Internet Explorer.

About this task

You can use these instructions if your Administrator sent you the appropriate files. Otherwise, follow the instructions in [Configuring and testing IE manually](#). On a 64-bit operating system, you must use 32-bit version of Internet Explorer. On Windows 2008, when Internet Explorer Enhanced Security Configuration for Users is ON, you must access Datacap Web Client from a different machine.

On	Do This
Workstation	Ensure that you add the tmweb.net address to IE as a trusted site.

On	Do This
Workstation	<p>In IE, on Tools > Internet Options > Security > Custom level</p> <p>Scroll down to Miscellaneous settings, enable Include local directory path when uploading files to a server.</p>
Workstation	<p>Configure IE.</p> <p>Extract the files your Administrator provided.</p> <p>Double-click the WebClientConfig.exe file.</p> <p>Enter or ensure that the URL is the web server IP address or name (http://WebServerName/).</p> <p>Click Configure, click OK, click Exit.</p>
Workstation	<p>Test IE.</p> <p>Start IE, enter the appropriate web server URL and test page (http://127.0.0.7/tmweb.net/ietest.aspx, http://WebServerName/tmweb.net/ietest.aspx, or https://WebServerName/tmweb.net/ietest.aspx).</p> <p>Click Yes to download Datacap TIFF Viewer, Thumbnails, and DataEdit Controls.</p> <p>Click Test - red Xs change to green check marks.</p>

Parent topic: [Remote workstation setup](#)

Configuring and testing IE manually

The checklist includes the tasks to configure and test Internet Explorer manually, if you do not have access to the Datacap Web Client Configuration tool.

About this task

If you do not have access to the Datacap Web Client Configuration tool, you can configure Internet Explorer manually.

On	Do This
----	---------

On	Do This
Developer workstation, Workstation	<p>Configure IE.</p> <p>Start Internet Explorer.</p> <p>Tools > Internet Options > Security tab.</p> <p>Select Trusted sites > click Sites button.</p> <p>On Trusted sites dialog, clear Require server verification, and add either the default IP address of the Datacap web server (http://127.0.0.1) or the URL of the web server as it is defined in your network's DNS (http://WebServerName or https://WebServerName).</p> <p>Security tab > Click Custom level button > Security settings dialog - Trusted Sites Zone:</p> <p>Enable: Download signed ActiveX controls and Initialize and script ActiveX controls not marked as safe for scripting.</p> <p>Scroll down to Miscellaneous settings > Enable Include local directory path when uploading files to a server.</p> <p>Click OK, then close IE.</p>
Developer workstation, Workstation	<p>Test IE.</p> <p>Start IE, enter the appropriate web server URL and test page (http://127.0.0.7/tmweb.net/ietest.aspx, http://WebServerName/tmweb.net/ietest.aspx, or https://WebServerName/tmweb.net/ietest.aspx).</p> <p>Click Yes to download Datacap TIFF Viewer, Thumbnails, and DataEdit Controls.</p> <p>Click Test. The red Xs change to green check marks.</p>

Parent topic: [Remote workstation setup](#)

Run the Datacap Web Client application

The checklist to run the Datacap Web Client application includes starting the Datacap server service and entering the appropriate web server URL address on your browser.

On	Do This
Datacap Server	<p>Start Datacap Server Service.</p> <p>In the Start menu, click IBM Datacap Services> Datacap Server Manager> Start</p>

On	Do This
Workstation	<p>Run Datacap Web Client.</p> <p>Start IE, and enter the appropriate web server URL (http://127.0.0.7/tmweb.net, http://WebServerName/tmweb.net, or https://WebServerName/tmweb.net).</p> <p>Sign on as Datacap User Admin, Password admin, and Station 1.</p> <p>Run and test the client using the images in the application's images folder. Confirm that each job in the workflow is processing correctly. You have to switch back to thick client to run some jobs.</p>

Parent topic: [Remote workstation setup](#)

User workstation and permissions setup

The checklist to set up the user workstation includes setting up sharing and security permissions, importing encryption keys, and installing the Datacap client components.

On	Do This
Workstation	<p>Ensure that Microsoft .NET Framework 3.5.1 is Installed</p> <p>Start > Control Panel > Programs > Programs and Features</p> <p>Select Turn Windows features on or off</p> <p>Select Microsoft .NET Framework 3.5.1 and click OK</p> <p>Reboot the machine, if prompted.</p>
Workstation	<p>Run installation program wizard.</p> <p>Install the Datacap client component - exclude all components except Datacap Client and the connectors to which you are entitled</p> <p>Expand Datacap client. Exclude the Applications, Datacap Studio, and Maintenance Manager components. Exclude FastDoc, if you are not using it.</p>
Workstation	<p>Import encryption keys</p> <p>Navigate to the Datacap server on which you generated keys</p> <p>Copy the dc_KTF.xml file from Datacap server to the C:\Datacap\Taskmaster folder on the developer workstation.</p>

On	Do This
Datacap server	<p>Set up sharing and security permissions for users.</p> <p>Set up sharing on the Datacap folder. Add or ensure that the domain/Windows user IDs of Datacap users are set to allow Full Control.</p> <p>Set up security on Datacap folder: Add or ensure that the domain/Windows user IDs of Datacap users are set to allow Read & Execute.</p> <p>Set up security on the application folder. Add or ensure that the domain/Windows user IDs of Datacap users are set to allow Full Control</p>
Workstation	<p>Optional: Install Scanner</p> <p>Ensure that scanner works outside of Datacap, set up application to use scanner</p>
Workstation	<p>Set the location of Datacap.xml file.</p> <p>In the Start menu, click IBM Datacap Services > Datacap Application Manager</p> <p>Click Service tab. Ensure that path location of the datacap.xml on Datacap server is correct.</p>

Parent topic: [Client/server installation checklist](#)

Datacap Report Viewer setup

To install Report Viewer in a client server environment, you must perform the required set up tasks on the server and the workstation.

On	Do This
Server	<p>Ensure that a domain/Windows Account Exists for Report Viewer</p> <p>When Datacap Web or Report Viewer are installed on the same web server, they must use the same domain/Windows account.</p> <p>Ensure that a domain/Windows account exists for Report Viewer</p>
Web server	<p>Add Report Viewer account to Administrators group</p> <p>Add domain/Windows account to Administrators group</p>
Server	<p>Stop Datacap Server Service</p> <p>In the Start menu, click IBM Datacap Services> Datacap Server ManagerStop</p>

On	Do This
Web server	<p>Ensure that Microsoft .NET Framework 3.5.1 is Installed</p> <p>Start > Administrative Tools > Server Manager</p> <p>In the Server Manager hierarchy pane, select Features</p> <p>Click Add Features</p> <p>Select .NET Framework 3.5.1 Feature, click Next</p> <p>Click Install</p>
Web server	<p>Ensure that Required IIS Components are Installed</p> <p>Start Administrative Tools > Server Manager</p> <p>Expand Roles, select Web Server (IIS)</p> <p>In Web Server (IIS) pane:</p> <p>Under Common HTTP Features, ensure Static Content is installed</p> <p>Scroll down to Role Services, under Application Development ensure that ASP.NET and ASP modules are installed</p> <p>Scroll down further to Management Tools, ensure that IIS Management Console is installed</p>
Server	<p>Set Report Viewer Sharing Permissions on Datacap Folder</p> <p>Right-click C:\Datacap folder, select Sharing, select Advanced Sharing</p> <p>Click Permissions, add Report Viewer account, set to allow Read</p>
Server	<p>Set Report Viewer Security on Datacap Folder</p> <p>Right-click C:\Datacap folder, select Properties</p> <p>Click Security tab</p> <p>Add Report Viewer account, set to allow Read & Execute</p>
Web server	<p>Install Report Viewer on web server</p> <p>Install only Report Viewer component</p>
Web server	<p>To use ADSI or LDAP authentication with Report Viewer, confirm that <code>EnableLDAP value="true"</code> in the Datacap\RV2\web.config file.</p>
Web server	<p>Import encryption keys</p> <p>Navigate to the Datacap Server on which you generated keys</p> <p>Copy the dc_KTF.xml file from Datacap Server to the C:\Datacap\Taskmaster folder on the web server</p>

On	Do This
Web server	<p>Add Application Pool for Report Viewer</p> <p>Start > Administrative Tools > Internet Information Services Manager</p> <p>Connections pane > Expand the Server > right-click Application Pools, select Add Application Pool</p> <p>Set: Name to Report Viewer</p> <p>Set: .NET Framework version to v 4.0.30319</p> <p>Set: Managed pipeline mode to Integrated</p> <p>Select: Start application pool immediately, click OK</p>
Web server	<p>Create Report Viewer Web Site</p> <p>Start > Administrative Tools > Internet Information Services Manager</p> <p>Connections pane > Expand the Server > expand Sites</p> <p>Right-click Default Web Site, select Add Application</p> <p>Set: Alias to Report Viewer</p> <p>Click Select, select the Report Viewer Application Pool you added (Report Viewer), click OK</p> <p>Set: Physical path to C:\Datacap\RV2, click OK</p> <p>Connections pane, select Application Pools</p> <p>Select Report Viewer application pool, then in Actions pane, click Advanced Settings</p> <p>Ensure that Microsoft .NET version is set to v4.0, ensure that Enable 32-Bit Applications is set to True</p> <p>Click Browse next to Process Model Identity, select Custom account, click Set and enter Report Viewer domain/Windows account information (accountname@domainname), click OK</p> <p>Set Load User Profile to True, click OK</p> <p>Select the Report Viewer site, double-click Session State</p> <p>Under Cookie Settings, change Name to Report Viewer, click Apply</p> <p>Connections pane, select Default Web Site, in Actions pane, under Manage Web Site, click Restart</p> <p>Ensure that Web Server, Application Pool and Default Web Site are all started</p>
Web server	<p>Set Location of Datacap.xml</p> <p>In the Start menu, click IBM Datacap Services and select Datacap Application Manager.</p> <p>Click Service tab, change path to reflect the location where the datacap.xml file is located on the Server (\\Server\Datacap\datacap.xml)</p>

On	Do This
Web server	<p>Set the database type</p> <p>Open the \Datacap\RV2 folder.</p> <p>Back up the reports.xml file and then edit the original file by setting the dbtype parameter:</p> <p>Microsoft Access <code>dbtype="0"</code></p> <p>SQL Server <code>dbtype="1"</code></p> <p>Oracle <code>dbtype="2"</code></p> <p>DB2 <code>dbtype="3"</code></p> <p>Apply for all applicable reports, and then save reports.xml.</p>

- [Viewing Datacap Report Viewer reports](#)

To view Report Viewer reports in a client server environment, you must perform the required set up tasks on the server and the workstation.

Parent topic: [Client/server installation checklist](#)

Viewing Datacap Report Viewer reports

To view Report Viewer reports in a client server environment, you must perform the required set up tasks on the server and the workstation.

On	Do This
Server	<p>Start Datacap Server Service</p> <p>Start > All Programs > Datacap > Administrator > Datacap > Datacap Server Manager > Start</p>
Workstation or developer workstation	<p>Add Report Viewer Address as Trusted Site</p> <p>Start IE > Tools > Internet Options > Security tab > Select Trusted sites > Click Sites button > Trusted sites dialog</p> <p>On Trusted sites dialog, if Report Viewer URL starts with http:, clear Require server verification, and add either the default IP address of the Datacap web server (http://localhost) or the web server's IP address or name as it is defined in your network's DNS (http://WebServerName or https://WebServerName), click Add</p> <p>Click Close</p>
Workstation or developer workstation	<p>Log onto Report Viewer, View Standard Report</p> <p>Start IE, enter http://WebServerName/RV2/Login.aspx</p> <p>Select Batch Productivity report, select application, click Run Report</p>

On	Do This
Workstation or developer workstation	<p>Create Report Viewer Report Filter</p> <p>On home page, select report</p> <p>Click Manage Filters</p> <p>In Add Filter field, enter name for new filter, click Add</p> <p>Select column name, select action, enter value</p> <p>Click Add Field to add another field</p> <p>Select Public or Private</p> <p>Click Save</p> <p>Click Run Report</p>
Workstation or developer workstation	<p>Add Reports to Report Viewer Dashboard</p> <p>On home page, click Dashboard</p> <p>Select report and application name, select filter</p> <p>Click Add to add another report</p> <p>Click Refresh and select a refresh interval</p>

Parent topic: [Datacap Report Viewer setup](#)

Rulerunner installation and configuration

The checklist includes tasks to set security and sharing permissions on the Datacap\RRS folder, install Rulerunner, and import encryption keys.

On	Do This
All	<p>Shut down Datacap software on all machines.</p> <ul style="list-style-type: none"> • Datacap client software. • Datacap Web Client. • Datacap Web Services • Datacap Server service. •
Datacap server	<p>Set sharing permissions on the Datacap folder.</p> <ul style="list-style-type: none"> • Right-click the C:\Datacap folder and select Properties. • Click the Sharing tab and click Advanced Sharing. • Click Permissions. • Add or ensure that the domain/Windows account for Rulerunner is set to allow Full Control.

On	Do This
Datacap server	Set up security on Datacap folder. <ul style="list-style-type: none"> • Right-click the C:\Datacap folder and select Properties. • Click the Security tab and click Edit. • Add or ensure that the domain/Windows account for Rulerunner set to allow Full Control.
Datacap server	Set up security on the Datacap\RRS folder. <ul style="list-style-type: none"> • Right-click C:\Datacap\RRS folder and select Properties. • Click Security tab and click Edit. • Add or ensure that domain/Windows account for Rulerunner set to allow Full Control.
Datacap server	Set up security on Datacap\Application folder. <ul style="list-style-type: none"> • Right-click the C:\Datacap\Application folder and select Properties. • Click Security tab and click Edit. • Add or ensure that domain/Windows account for Rulerunner set to allow Full Control.
Rulerunner server	Ensure that Microsoft .NET Framework 3.5.1 is installed. <ul style="list-style-type: none"> • Start > Administrative Tools > Server Manager. • In the Server Manager hierarchy pane, select Features. • Click Add Features. • Select .NET Framework 3.5.1 Feature and click Next. • Click Install.
Rulerunner server	Install Rulerunner on the Rulerunner server. <ul style="list-style-type: none"> • Install Datacap client, Rulerunner, and separately licensed connectors to which you are entitled.
Rulerunner server	Install all required third-party software components, if Rulerunner is exporting to IBM® or third-party repositories.
Rulerunner server	Import encryption keys. <ul style="list-style-type: none"> • Navigate to the Datacap server on which you generated keys. • Copy the dc_KTF.xml file from the Datacap Server to the C:\Datacap\Taskmaster folder on the Rulerunner server.
Rulerunner server	Set the location of Datacap.xml file. <ul style="list-style-type: none"> • In the Start menu, click IBM Datacap Services > Datacap Application Manager. • Click the Service tab and change the path to the location of the datacap.xml file, such as \\Server\Datacap\datacap.xml.

- [Configure Rulerunner account permissions](#)

The checklist to configure the Rulerunner account includes tasks for setting up security and granting

permissions.

- [Configure Rulerunner authentication](#)

The checklist includes tasks that setting up Rulerunner to authenticate with the Datacap authentication method or an external authentication system.

- [Configure Rulerunner to run application tasks](#)

The checklist includes configuring task profiles in the Datacap Application Manager and adding threads for the Rulerunner to run application tasks.

Parent topic: [Client/server installation checklist](#)

Configure Rulerunner account permissions

The checklist to configure the Rulerunner account includes tasks for setting up security and granting permissions.

On	Do This
Rulerunner server	<p>Grant Rulerunner account DCOM Config permissions.</p> <p>Start > Administrative Tools > Component Services > Computers > My Computer > DCOM Config > DCOProcessor application.</p> <p>Right-click DCOProcessor application and select Properties.</p> <ul style="list-style-type: none">• Click the Security tab. Set Launch and Activate permissions to Customize and click Edit.• Add Rulerunner account and set Local Launch and Local Activation to Allow. <p>Right-click RRProcessor and select Properties.</p> <ul style="list-style-type: none">• Click the Security tab. Set Launch and Activate permissions to Customize and click Edit.• Add Rulerunner account and set Local Launch and Local Activation to Allow.
Rulerunner server	<p>Set up security on systemprofile\AppData folder.</p> <ul style="list-style-type: none">• Right-click C:\Windows\SysWOW64\config\systemprofile\AppData folder and select Properties.• Click Security tab, click Edit.• Add or ensure that domain/Windows account for Rulerunner set to allow Modify.
Rulerunner server	<p>Grant Rulerunner Log On as a Service privilege.</p> <ul style="list-style-type: none">• Start > Administrative Tools > Services.• Right-click Rulerunner Service and select Properties.• Click the Log On tab and select This account. Click Browse and select the Rulerunner account and enter the password. Click Apply.

Parent topic: [Rulerunner installation and configuration](#)

Configure Rulerunner authentication

The checklist includes tasks that setting up Rulerunner to authenticate with the Datacap authentication method or an external authentication system.

On	Do This
Datacap server	Create or ensure that a domain/Windows account exists for Rulerunner. All instances of Rulerunner can use the same account.
Datacap server	Start the Datacap Server service. In the Start menu, click IBM Datacap Services > Datacap Server Manager .
Developer workstation	Log in to Datacap Web Client. <ul style="list-style-type: none"> • Open Internet Explorer and enter http://WebServerName. • Select application, enter user ID, password, station, and click Login.
Developer workstation	When you are using Datacap authentication <ul style="list-style-type: none"> • Create or ensure a Datacap user exists that Rulerunner can use. • Create or ensure a Datacap station exists that Rulerunner can use. • Set up Rulerunner to use Datacap Authentication and enter Datacap user, password, and station.
Developer workstation	When you are using ADSI or LDAP <ul style="list-style-type: none"> • Obtain the name of the domain and authentication security group. • Create or ensure a domain/Windows account exists that Rulerunner can use. • Add group to Datacap application where Name is a group ID that is a concatenation of your security group name, a dot, and the short domain name. • Add station to Datacap application where Name is the name of Rulerunner server, and set Maximum to 9999. • Set up Rulerunner to use Windows Authentication.
Developer workstation	When you are using ADLDS or LLDAP <ul style="list-style-type: none"> • Create or ensure that an account for Rulerunner is set up in ADLDS or LLDAP authentication system. • Add Datacap user to application with same name as authentication system account. • Add station to Datacap application. • Set up Rulerunner to use Datacap Authentication and enter Datacap user, password, and station.

Parent topic: [Rulerunner installation and configuration](#)

Configure Rulerunner to run application tasks

The checklist includes configuring task profiles in the Datacap Application Manager and adding threads for the Rulerunner to run application tasks.

On	Do This
----	---------

On	Do This
Various	<p>Ensure the following before starting.</p> <ul style="list-style-type: none"> • Ensure that you have working versions of your Datacap applications and that you have successfully run all tasks manually. • Configure Rulerunner authentication. • Install Rulerunner on a server that is running Windows Server 2008. • Determine which Datacap tasks to process, and gather required task information.
Developer workstation	<p>Configuring task profiles in the Datacap Application Manager.</p> <ul style="list-style-type: none"> • In the Start menu, click IBM Datacap Services>Datacap Application Manager • Select application, ensure that paths are correct. • Click Rulerunner tab. Click the red X to remove task profiles that Rulerunner is not to process. • Click Add new Task to add a profile that Rulerunner is to process. Profile names are case-sensitive.
Various	<p>Stop and restart Datacap server service.</p> <ul style="list-style-type: none"> • In the Start menu, click IBM Datacap Services>Datacap Server Manager. • Click Stop. • Click Start.
Rulerunner server	<p>Start Rulerunner.</p> <ul style="list-style-type: none"> • In the Start menu, click IBM Datacap Services>Datacap Rulerunner Manager > Start. • Click Rulerunner Login . Select Datacap Authentication and enter credentials. Click Connect. • Click Workflow:Job:Task tab. Click the box next to MyApp1 to expand the tree.
Rulerunner server	<p>Configure Rulerunner to run a task.</p> <ul style="list-style-type: none"> • Right-click Threads. Select Add Thread to add a single thread or Add Threads to add more than one thread. • In the other pane, under MyApp1, click the box next to the node. Click and drag the node onto the thread in right pane. • Click Save and Yes. • If required, click node and change priority or increase value for skipsamebatch. • If required, click the Logging tab and change the settings. • Click Save. • Click Rulerunner Login tab, click Disconnect.

On	Do This
Rulerunner server	Reset Rulerunner Login Credentials. Reset login credentials for Rulerunner, if you are not using Datacap Authentication.
Rulerunner server	Run <code>Myapp1</code> tasks, start Rulerunner, and monitor batches that are processed by Rulerunner.

Parent topic: [Rulerunner installation and configuration](#)

Fingerprint Service setup

The checklist includes tasks to configure the Fingerprint service application pool and set up security on the Datacap and application folders.

On	Do This
Datacap server	Create or ensure that a domain/Windows account exists for the Datacap Fingerprint service.
Datacap server	Set Sharing Permissions for Datacap folder. <ul style="list-style-type: none"> Right-click C:\Datacap folder and select Properties. Click Sharing tab, click Advanced Sharing. Click Permissions. Add or ensure that domain/Windows account for Fingerprint Service set to allow Full Control.
Datacap server	Set Security on Datacap\Application\fingerprint folder. <ul style="list-style-type: none"> Right-click C:\Datacap\Application\fingerprint folder and select Properties. Click Security tab, click Edit. Add or ensure that domain/Windows account for Fingerprint Service is set to allow Read & Execute.
Datacap server	Set Security on Datacap\Application\Batches folder. <ul style="list-style-type: none"> Right-click C:\Datacap\Application\Batches folder and select Properties. Click Security tab, click Edit. Add or ensure that domain/Windows account for Fingerprint Service is set to allow Read & Execute.
Fingerprint server	Install Fingerprint service. Install only Rulerunner component.

On	Do This
Fingerprint server	Set Security on Datacap\FingerprintService folder. <ul style="list-style-type: none"> • Right-click C:\Datacap\FingerprintService folder and select Properties. • Click Security tab, click Edit. • Add NETWORK SERVICE and local IUSR and set both to allow Read & Execute.
Fingerprint server	Add Fingerprint Service account to IIS_IUSRS group. <ul style="list-style-type: none"> • Start > Control Panel > Administrative Tools > Computer Management > Local Users and Groups > Groups > IIS_IUSRS and select Properties. • Add Fingerprint Service domain/Windows account, click OK.
Fingerprint server	Add Application Pool for Fingerprint service. <ul style="list-style-type: none"> • Start > Control Panel > Administrative Tools > Internet Information Services Manager > Connections pane. Expand the server name and right-click Application Pools. Select Add Application Pool. • Set Name to <i>fp</i>service. • Set .NET Framework version to <i>v4.0.30319</i>. • Set Managed pipeline mode to <i>Integrated</i>. • Select Start application pool immediately,
Fingerprint server	Set up Fingerprint service. <ul style="list-style-type: none"> • Start > Control Panel > Administrative Tools > Internet Information Services Manager > Connections pane. Expand Server and Sites. Right-click Default Web Site and select Add Application. • Set Alias to <i>fp</i>service. • Click Select, select <i>fp</i>service Application Pool, click OK. • Set Physical path to C:\Datacap\FingerprintService. • Click OK. • Click Test Settings, click Close, click OK. • In Connections pane, select Application Pools • Select <i>fp</i>service Application Pool and click Advanced Settings. • Set Enable 32-Bit Applications to <i>True</i>. • In Process Model > Identity, click the browse button. Select Custom account and Set. • Enter the Fingerprint Service domain/Windows account information in the format, <i>accountname@domainname</i>. This information is the same account that you added to the Fingerprint server IIS_IUSRS group. Enter the account's password twice, click OK • In Process Model > Idle Time-out, set to zero, click OK.

- [Test Fingerprint Service setup](#)

The checklist includes tasks to validate the Fingerprint service installation and confirm the service can load fingerprints.

Parent topic: [Client/server installation checklist](#)

Test Fingerprint Service setup

The checklist includes tasks to validate the Fingerprint service installation and confirm the service can load fingerprints.

On	Do This
Fingerprint server	Validate Fingerprint Service Installation. <ul style="list-style-type: none">• Start Internet Explorer enter <code>http://127.0.0.1/fpservice/Service.asmx?WSDL</code> to display the Service CCO Fingerprints DB Service page.
Fingerprint server	Confirm the Fingerprint service can load fingerprints. <ul style="list-style-type: none">• In the Start menu, click IBM Datacap Developer Tools>Datacap Fingerprint Service Test Tool .• Enter the application name.• Enter Fingerprints URL as <code>http://127.0.0.1/fpservice/Service.asmx?WSDL</code>.• Enter the full UNC path to the fingerprint directory of your application in the Fingerprint Directory field. Click Upload All Fingerprints From Directory, confirm number of Fingerprints Loaded is correct and the names are correct.• Confirm that the Fingerprint Service matches .cco file to a loaded fingerprint. Enter the full path of one fingerprint, which is displayed in the pane, in the Find Fingerprint field. Click Find Fingerprint. Match results should be <code>1.00;1;0;0</code>.

Parent topic: [Fingerprint Service setup](#)

Datacap Maintenance Manager setup

You must complete the required account and configuration settings before you can run Datacap Maintenance Manager on Datacap in a client/server environment. You can run Maintenance Manager for demonstration, proof of concept, development, and test purposes.

About this task

On	Do This
Server	Create or ensure that a domain/Windows account exists for Maintenance Manager Create or ensure that a domain/Windows account exists for the developer
Server	Ensure that the domain/Windows account for the developer was granted sharing and security permissions on the Server. These permissions allow the development of a new Maintenance Manager application, see Configuring Datacap on the server .

On	Do This
Developer workstation, the computer on which Maintenance Manager runs in production	Add Maintenance Manager domain/Windows account to the Administrators or Backup Operators group
Server	Set Maintenance Manager account sharing permissions for Datacap folder Right-click C:\Datacap folder and select Properties Click Sharing tab, click Advanced Sharing Click Permissions Add or ensure that Domain/Windows account for Maintenance Manager set to allow Full Control
Server	Set Maintenance Manager account security permissions for Datacap folder Right-click C:\Datacap folder and select Properties Click Security tab Add or ensure that Domain/Windows account for Maintenance Manager set to allow Read & Execute
Server	Set Maintenance Manager account security permissions for Datacap\RRS folder Right-click C:\Datacap\RRS folder and select Properties Click Security tab Add or ensure that Domain/Windows account for Maintenance Manager set to allow Read & Execute
Developer workstation	Run installation program wizard Install Datacap Client component (including Datacap Studio, Maintenance Manager), and the applications and connectors to which you are entitled
Developer workstation	Import encryption keys Go to the Datacap Server on which you generated keys Copy the dc_KTF.xml file from Datacap Server to the C:\Datacap\Taskmaster folder on the developer workstation

Parent topic: [Client/server installation checklist](#)

Datacap installation command-line parameters

You can run the Datacap installation program Setup.exe file by using command-line parameters from the command line.

About this task

The parameters that are available are the installation program parameters, the Microsoft Windows Installer parameters, and the Datacap-specific parameters.

- [Viewing the installation program parameters](#)
Open the Datacap installation program Setup.exe file to display a list of the parameters.
- [Microsoft Windows Installer parameters](#)
Open the msiexec.exe file in Datacap to display a list of the Microsoft Windows Installer parameters.
- [Commonly used Datacap Setup.exe parameters](#)
You can use the Datacap Setup.exe parameters to run a silent installation of Datacap and many of its components.

Parent topic: [Installing and configuring in a client/server environment](#)

Viewing the installation program parameters

Open the Datacap installation program Setup.exe file to display a list of the parameters.

About this task

To view the installation program parameters that are available, follow this procedure.

Procedure

1. From the Windows Start menu, select Run. The Run dialog opens.
2. Click Browse and go to the folder that contains the Datacap installation program Setup.exe file.
3. Select Setup.exe and click Open. The Browse window closes and the full path to the Setup.exe file is displayed in the Run dialog.
4. After Setup.exe, add a space and /? then click OK. The installation program parameters are displayed.
5. Click OK to close the installation program message box.

Parent topic: [Datacap installation command-line parameters](#)

Microsoft Windows Installer parameters

Open the msiexec.exe file in Datacap to display a list of the Microsoft Windows Installer parameters.

About this task

To view the Microsoft Windows Installer parameters that are available:

Procedure

1. From the Windows Start menu, select Run. The Run dialog opens.
2. Click Browse and navigate to the folder containing the Microsoft Windows Installer program msiexec.exe file.
3. Select msiexec.exe and click Open. The Browse window closes and the full path to the msiexec.exe file is displayed in the Run dialog.
4. After msiexec.exe, add a space and /? (for example: C:\WINDOWS\system32\msiexec.exe /?), then click OK. The Windows Installer parameters are displayed.
5. Click OK to close the Windows Installer message box.

Parent topic: [Datacap installation command-line parameters](#)

Commonly used Datacap Setup.exe parameters

You can use the Datacap Setup.exe parameters to run a silent installation of Datacap and many of its components.

The following table describes how to use the command lines that are most commonly used when you are installing Datacap.

Attention:

- When these commands are run, they must be run as Administrator.
- When you are installing components individually, you must also install the LAP and Shared components, otherwise the individual component does not function properly.
- Name/value pairs, such as `INSTALLDIR=\"<path>\"` and `ADDDEFAULT=LAP, Shared, Server`, are case-sensitive. Each Name is capitalized. The case values are as shown in the table.
- Replace `<path>` with full path (for example `C:\Datacap`).
- Replace `<filename>` with name to assign to log file (for example: `InstallLog`).
- To prevent unintended changes to the main Datacap application settings file (`datacap.xml`), the version of the Application Manager that is installed with every component during a command-line installation is a restricted version. The restricted version allows only the location of the `datacap.xml` file to be changed. Add the `DSACTION=3` parameter to the command to install the unrestricted version of the Application Manager. For an example, see the reference in the following table, *Datacap Desktop and Datacap Studio*.

To Do This	Use This Command Line
Silent Installation: Quiet mode, no user interaction, all components installed	<code>setup.exe /S /V"/quiet DSACTION=3"</code>
Silent Installation: Quiet mode, different location, all components installed	<code>setup.exe /S /V"/quiet INSTALLDIR=\"<path>\" DSACTION=3"</code>
Silent Installation: Unattended mode, progress bar only, all components installed	<code>setup.exe /S /V"/passive DSACTION=3"</code>
Silent Installation: Unattended mode, different location, all components installed	<code>setup.exe /S /V"/passive INSTALLDIR=\"<path>\" DSACTION=3"</code>
Create Install Log, all components installed	<code>setup.exe /V"/log <path>\<filename>.txt DSACTION=3"</code>
Install Datacap Server only	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Server"</code>
Install Datacap Server and Datacap Web Client	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Server,Web"</code>
Install Datacap Desktop	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Client"</code>
Install Datacap Desktop and Datacap Accounts Payable application	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Client,APT"</code>
Install Datacap Desktop and Medical Claims Application	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Client,Mclaims"</code>
Install Datacap Desktop and FastDoc	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Client,FastDoc"</code>
Install Datacap Desktop and Datacap Studio	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Client,DStudio DSACTION=3"</code>
Install Datacap Desktop and Maintenance Manager (Notification)	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Client,NEU"</code>

To Do This	Use This Command Line
Install Rulerunner (Rulerunner Service and Fingerprint Service)	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Rulerunner"</code>
Install Datacap Web Client	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Web"</code>
Install Datacap Web Services	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,wTM"</code>
Install Report Viewer	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,RV2"</code>
Install eMail Input (eMail & eDoc)	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,eMail"</code>
Install Fax Input	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Fax"</code>
Install Documentum Connector	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,Documentum"</code>
Install SharePoint Connector	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,SharePoint"</code>
Install Insight Edition (Document Analytics)	<code>setup.exe /S /V"/passive ADDDEFAULT=LAP,Shared,DocumentAnalytics"</code>

Parent topic: [Datacap installation command-line parameters](#)

Configuring Datacap databases

The Datacap applications can use DB2®, Microsoft SQL Server, or Oracle databases to store workflow definitions, security parameters, processing information, and fingerprints.

Datacap is installed with sample applications and templates that use Microsoft Access for their application databases. However, Microsoft Access is not supported in a Datacap production environment.

You can configure the following Datacap application databases to use DB2, Microsoft SQL Server, or Oracle:

Administration database

Contains definitions of the workflows, jobs, and tasks in the application. Also contains the security parameters for the users, user groups, stations, and job-task shortcut icons in the application.

Engine database

Stores current and historical processing information for each batch of documents and its contents.

Fingerprint database

Manages the fingerprints of the application if you are using matching fingerprints to identify and align forms for the application.

Note: For the Admin, Engine, and Fingerprint databases, when connecting to an Access database, use Microsoft Access (Jet). If Microsoft Access (Jet) interface is not used, then the applications might go through connection issues. Microsoft Access (ACE) interface is generally used with a specific type of databases such as LookupDB, and not recommended to connect with Admin, Engine, or Fingerprint databases.

Lookup and other application-specific databases that do database tasks for the Datacap applications can also be configured to use DB2, Microsoft SQL Server, or Oracle.

The following information describes how to define the database structure, populate the database, move the database, and configure the application to use the database. This information does not describe how to do lookups or export to the database.

Prerequisites

Before you can configure a DB2, SQL Server, or Oracle database for a Datacap application, you must have access to the database. A Database Administrator must initially create each DB2, SQL Server, or Oracle database and grant appropriate access rights.

For SQL Server, if you want to create an isolated test environment, you can use Microsoft SQL Server Express®, which you download from the Microsoft website.

For Oracle, the Oracle client must be installed on any computers that communicate directly with an Oracle database. You must install the appropriate third-party drivers or software (for example, Oracle connectivity software) on the computer where a Datacap component needs to communicate with the database. For example, the third-party drivers or connectivity software must be installed on:

- The machine where Datacap Server is installed.
- The machine where Rulerunner is installed, if an application calls actions that performs lookups to these databases.
- Other machines running the client software if the clients perform lookups or other queries to the database.

The computers on which you run the clients that connect to the Fingerprint database or Oracle, or run rules that make SQL calls to any Oracle database, such as for lookups, require the Oracle client installed. These clients include Datacap Desktop, Datacap FastDoc, Rulerunner, and Datacap Web Client. When you are using the Application Copy Tool, that computer must have the Oracle client that is installed and connectivity to any source or destination Oracle database.

For DB2, the DB2 client must be installed on any computers that communicate directly with a DB2 database. You must install the appropriate third-party drivers or software (for example, DB2 connectivity software) on the computer where a Datacap component needs to communicate with the database. For example, the third-party drivers or connectivity software must be installed on:

- The machine where Datacap Server is installed.
- The machine where Rulerunner is installed, if an application calls actions that performs lookups to these databases.
- Other machines running the client software if the clients perform lookups or other queries to the database.

The computers on which you run the clients that connect to the Fingerprint database or DB2, or run rules that make SQL calls to any DB2 database, such as for lookups, require the DB2 client installed. These clients include Datacap Desktop, Datacap FastDoc, Rulerunner, and Datacap Web Client. When you are using the Application Copy Tool, that computer must have the DB2 client that is installed and connectivity to any source or destination DB2 database.

- [Configuring an application to use the database](#)
You configure the Datacap application for a DB2, Microsoft SQL Server, or Oracle database by using Datacap Application Manager to set the encrypted connection strings that are stored in the application configuration (.app) file.
- [Database security permissions](#)
Datacap users must have database permissions for DB2, Microsoft SQL Server, or Oracle to create the database and its schema, initialize the data in the database, and run daily operations on the database.
- [Defining the database structure](#)
SQL scripts define the structure of the Datacap application databases for DB2, Microsoft SQL Server, or Oracle databases.
- [Moving the application files](#)
You can use the Datacap Application wizard to copy a Microsoft Access based application and move it to a folder on another computer.

- [Verifying the database connection](#)

After you configure the application for the database, verify the connection to the DB2, Microsoft SQL Server or Oracle database works properly.

Configuring an application to use the database

You configure the Datacap application for a DB2®, Microsoft SQL Server, or Oracle database by using Datacap Application Manager to set the encrypted connection strings that are stored in the application configuration (.app) file.

About this task

You use the Datacap Application Manager to configure the application to use the database. Datacap Application Manager works well with separate Test and Production environments because it stores the information that is specific to the Datacap application environment such as database locations and physical paths.

In a Test and Production paradigm, the physical applications are identical. The rules, workflows, page types, and other components, are identical, except for the *.app file. The difference between the environments is what the Datacap Application Manager stores. In the Test environment, the Datacap Application Manager stores the Test environment details while the Production system has its own unique environmental settings.

When you configure Datacap for a DB2, Microsoft SQL Server, or Oracle database, you can put the Administration, Engine, and Fingerprint schemas for the same application in a single database. You cannot put schemas for different applications in the same database.

For Oracle, each computer that accesses the database must have the Oracle 32-bit client software installed. A net service name must be created on that computer as well.

For DB2, if the user ID that created the database tables is different from user ID that is used in the connection string, you must also specify the user ID that created the database tables as the schema. To specify the correct schema, use the user ID in the connection string. Use *CurrentSchema=user ID that created the tables*.

Procedure

To configure an application to use the database:

1. In the Start menu click IBM Datacap Services > Datacap Application Manager.
2. Select a Datacap application. For example, select TravelDocs.
3. Click the Main tab.
4. In the Administration field, click Browse and select a database provider.
5. Enter the database authentication information and click OK.
6. In the Engine field, repeat the Administration database steps.
7. Close the Datacap Application Manager.
8. If you are using a Fingerprint or Lookup database, scroll down to the Workflows section and repeat these steps for each database.

Parent topic: [Configuring Datacap databases](#)

Database security permissions

Datacap users must have database permissions for DB2®, Microsoft SQL Server, or Oracle to create the database and its schema, initialize the data in the database, and run daily operations on the database.

You set up database permissions for Datacap users by using the database-specific tool and following the instructions that are provided with your database:

- For DB2, contains multiple tools that you can use to configure DB2 databases. Select one of the following tools and log on as administrator:
 - Datacap Studio: the Datacap application development tool
 - DB2 Command Line Processor CLP: a command line equivalent to Oracle SQLPlus
 - DB2 Command Line Processor CLPlus: another command-line equivalents to Oracle SQLPlus
- For SQL Server, start SQL Server Management Studio or SQL Server Enterprise Manager and log on to the server as the administrator
- For Oracle, start the SQLPlus tool and log on to the SYS account as SYSDBA

Assign the appropriate permissions from the following table to the Datacap users based on these tasks:

- Database administrators: create the database and its schema
- Datacap developers: initialize the data in the database
- Datacap Server: Run daily operations on the Administrator and Engine database tables
- Rulerunner and Fingerprint Services: Run daily operations on the Fingerprint database
- Datacap Web Client, Rulerunner, Datacap Desktop, and other thick clients: Run daily operations on queries that are run by actions. The permissions are application-specific and apply only to those queries that are run by actions. These users usually need read and in some cases write permissions to the Fingerprint database and any Lookup and Export databases that exist.

The following table lists the database permission and indicates which permissions are required for each of the database tasks.

Table 1. Required database permissions for SQL Server, Oracle, and DB2 database tasks

Database permission	Create database and schema	Initialize data in the database	Run daily operations
Create an index	Required	Not required	Not required
Drop an index	Required	Not required	Not required
Create a sequence	Required	Required	Not required
Create and initialize columns	Required	Required	Not required
Drop a sequence	Required	Required	Not required
Select a sequence	Required	Required	Required
Create a table	Required	Not required	Not required
Drop a table	Required	Not required	Not required
Insert a table	Required	Required	Required
Select a table	Required	Required	Required
Update a table	Required	Required	Required
Delete records from a table	Required	Required	Not required
Create a trigger	Required	Not required	Not required
Create a view	Required	Not required	Not required

Parent topic: [Configuring Datacap databases](#)

Defining the database structure

SQL scripts define the structure of the Datacap application databases for DB2®, Microsoft SQL Server, or Oracle databases.

About this task

Use the SQL scripts that are installed as part of the Datacap installation to create an empty database structure. You create this database structure for the Datacap Administration, Engine, and Fingerprint databases.

Procedure

To define the database structure:

1. Go to C:\Datacap\support\DBScript and select the scripts for the database you want to create:

Option	Description
DB2	DB2_Adm_Base.sql
	DB2_Eng_Base.sql
	DB2_FP_Base.sql
Microsoft SQL Server	SQL_Adm_Base.sql
	SQL_Eng_Base.sql
	SQL_FP_Base.sql
Oracle	Oracle_Adm_Base.sql
	Oracle_Eng_Base.sql
	Oracle_FP_Base.sql

2. For DB2, start the DB2 CLPPlus tool and log on as the owner of the database schema.
3. For SQL Server, start SQL Server Management Studio or SQL Server Enterprise Manager and log on to the server as the administrator.
4. For Oracle, start the SQLPlus tool and log on as the owner of the database schema.
5. Create the database by following the instructions that are provided in the Microsoft SQL Server or Oracle documentation that was provided with your database.
6. Open the appropriate scripts in the tool and run them to define the structure of the Datacap application databases.

Parent topic: [Configuring Datacap databases](#)

Creating a single DB2 database for a Datacap application

You can create a single DB2® database to store the Administration, Engine, and Fingerprint databases for a Datacap application.

Before you begin

You must create the DB2 database and set up the appropriate permissions to the database. Review and complete the prerequisites for configuring Datacap databases.

About this task

You define the structure of the Datacap application databases by running SQL scripts for the Administration, Engine, and Fingerprint database schema.

You can run the SQL scripts for these Datacap application databases separately to create a separate DB2, SQL Server, or Oracle database for each of the application databases. Alternatively, you can run the SQL scripts together to create just one database that contains the Administration, Engine, and Fingerprint database schemas. This procedure creates a single DB2 database user for all of the Datacap application databases.

Procedure

To create a single DB2 user for the Datacap databases in your Datacap application:

1. Create a new user for the DB2 database, for example, `new_db_user`. Grant DBADM privileges to `new_db_user`.
Important: The user name `new_db_user` is used as an example. Choose a more descriptive name such as `CaptureInvoice™` or `AcmeCapture` to identify the function of the particular Datacap application that this DB2 database will be used for.
2. Start the DB2 CLPPlus tool and log in as `new_db_user`.
3. Go to `C:\Datacap\support\DBScript` and find the following DB2 scripts:
 - o `DB2_Adm_Base.sql`
 - o `DB2_Eng_Base.sql`
 - o `DB2_FP_Base.sql`
 - a. If this is not the first database that you are creating in this Datacap environment, you must modify the scripts to have unique `db_index` values to uniquely associate with the application:

IDs for the `db_index` fields are hard-coded in the database creation scripts. If this is not the first database that you are creating in this Datacap environment, you must set a new numeric value in the `db_index` column in the scripts (the `adminfo` table in the Admin schema and the `enginfo` table in the Engine schema) to uniquely identify the database with this Datacap application.

If you have one admin database per engine database for an application, you can use the same unique numeric ID in both databases. If you have multiple admin databases per engine database, specify different unique numeric IDs for each database, but use a numbering convention that simplifies the identification of those databases for an application.

Tip: When batches are created in the Engine schema, each queue table row contains column `qu_admDB`. This value displays which admin database is associated with the batch.
 - b. Run the scripts.
4. Disconnect from the user account.
5. Copy the data from the source database by using the Datacap Application Copy Tool.
6. Point the Administration, Engine, and Fingerprint database schemas to the same location.
Tip: Because each of the Datacap databases is created in the same DB2 database, the connection string for each of the Datacap databases will be the same per Datacap application.
7. Configure the Datacap application to use the database by using the database user name that you created in this task for that database.

Related tasks:

[Configuring an application to use the database](#)

Related information:

[Authorities overview](#)

[Database administration authority \(DBADM\)](#)

Creating a single Oracle database for a Datacap application

You can create a single Oracle database for the Administration, Engine, and Fingerprint databases in a Datacap application.

About this task

You define the structure of the Datacap application databases by running SQL scripts for the Administration, Engine, and Fingerprint database schema. You can run SQL scripts for these Datacap application databases separately to create a SQL Server or Oracle database for each of the application databases.

For Oracle databases, you can run the SQL scripts together to create 1 Oracle database that contains the Administration, Engine, and Fingerprint database schema. This procedure creates a single Oracle database user for all the Datacap application databases.

Procedure

To create a single Oracle user for Datacap databases:

1. Start the SQLPlus tool and log on to the `SYS@net service name` as `SYSDBA`. The *net service name* is specific to each computer.
2. Run the following script to create the user:

```
CREATE USER APTOra3X
        IDENTIFIED BY APTOra3X;
```

```
Grant resource to APTOra3X;
Grant DBA to APTOra3X;
Grant EXP_FULL_DATABASE to APTOra3X;
```

Attention: The user name `APTOra3x` is used as an example. You can choose a more descriptive name such as `CaptureInvoiceTM` or `AcmeCapture` to identify the function of this particular Oracle database.

3. Disconnect from the `SYS` account and log on to the user account that you created with SQLPlus.
4. Go to `C:\Datacap\support\DBScript` and find the following Oracle scripts:
 - o `Oracle_Adm_Base.sql`
 - o `Oracle_Eng_Base.sql`
 - o `Oracle_FP_Base.sql`
 - a. If this is not the first database that you are creating in this Datacap environment, you must modify the scripts to have unique `db_index` values to uniquely associate with the application:

IDs for the `db_index` fields are hard-coded in the database creation scripts. If this is not the first database that you are creating in this Datacap environment, you must set a new numeric value in the `db_index` column in the scripts (the `adminfo` table in the Admin schema and the `enginfo` table in the Engine schema) to uniquely identify the database with this Datacap application.

If you have one admin database per engine database for an application, you can use the same unique numeric ID in both databases. If you have multiple admin databases per engine database, specify different unique numeric IDs for each database, but use a numbering convention that simplifies the identification of those databases for an application.

Tip: When batches are created in the Engine schema, each queue table row contains column `qu_admDB`. This value displays which admin database is associated with the batch.

- b. Run the scripts.
5. Disconnect from the user account.
 6. Copy the data from the source database by using the Application Copy Tool.

7. Point the Administration, Engine, and Fingerprint database schemas to the same location.
8. Configure the application to use the database by using the database user name that you created here.

Related tasks:

[Configuring an application to use the database](#)

Creating a single Microsoft SQL Server database for a Datacap application

You can create a single SQL Server database to store the Administration, Engine, and Fingerprint databases for a Datacap application.

Before you begin

You must create the SQL Server database and set up the appropriate permissions to the database. Review and complete the prerequisites for configuring Datacap databases.

About this task

You define the structure of the Datacap application databases by running SQL scripts for the Administration, Engine, and Fingerprint database schema.

You can run the SQL scripts for these Datacap application databases separately to create a separate DB2®, SQL Server, or Oracle database for each of the application databases. Alternatively, you can run the SQL scripts together to create just one database that contains the Administration, Engine, and Fingerprint database schemas. This procedure creates a single SQL Server database user for all of the Datacap application databases.

Procedure

To create a single SQL Server user for the Datacap databases in your Datacap application:

1. Start SQL Server Management Studio or SQL Server Enterprise Manager and log on to the server as the administrator.
2. Select the database that you created for your Datacap application.
3. Create a new login name (user) and password and assign the `db_owner` role to this new user. For more information about creating a new login, see the Microsoft SQL Server documentation for your database version.
4. Log on to the database with your new user name.
5. Go to `C:\Datacap\support\DBScript` and find the following SQL Server scripts:
 - o `SQL_Adm_Base.sql`
 - o `SQL_Eng_Base.sql`
 - o `SQL_FP_Base.sql`
 - a. If this is not the first database that you are creating in this Datacap environment, you must modify the scripts to have unique `db_index` values to uniquely associate with the application:

IDs for the `db_index` fields are hard-coded in the database creation scripts. If this is not the first database that you are creating in this Datacap environment, you must set a new numeric value in the `db_index` column in the scripts (the `adminfo` table in the Admin schema and the `enginfo` table in the Engine schema) to uniquely identify the database with this Datacap application.

If you have one admin database per engine database for an application, you can use the same unique numeric ID in both databases. If you have multiple admin databases per engine database,

specify different unique numeric IDs for each database, but use a numbering convention that simplifies the identification of those databases for an application.

Tip: When batches are created in the Engine schema, each queue table row contains column `qu_admDB`. This value displays which admin database is associated with the batch.

b. Run the scripts.

6. Disconnect from the user account.
7. Copy the data from the source database by using the Datacap Application Copy Tool.
8. Point the Administration, Engine, and Fingerprint database schemas to the same location.
Tip: Because each of the Datacap databases is created in the same SQL Server database, the connection string for each of the Datacap databases will be the same per Datacap application.
9. Configure the Datacap application to use the database by using the database user name that you created in this task for that database.

Related tasks:

[Configuring an application to use the database](#)

Converting applications from Jet database to Oracle database

If you have multiple applications that require Oracle database, you can convert the applications from a Microsoft Jet Database to a single Oracle database using Datacap Application copy tool.

About this task

Multiple Datacap applications within a single Oracle instance are differentiated by the user credentials used to access them. You can use the same Oracle database for multiple Datacap applications, as long as each unique Oracle "User" owns the schema for each application.

Note: You cannot combine schemas for multiple Datacap applications into a single Oracle schema.

Procedure

1. Create a unique database user for each application. This will create a schema for that user.

Note: You can optionally specify a new tablespace for each user/schema. If the tablespace is not specified, by default the users are created in the system tablespace. The users can be in the same tablespace or different tablespaces.

Repeat the following steps for each user.

2. Log on to Oracle as one of the users created in step 1.
3. Navigate to ...\`Datacap\support\DBScript\` and run the following Oracle scripts:
 - o `Oracle_Adm_base.sql`
 - o `Oracle_Eng_base.sql`
 - o `Oracle_FP_base.sql`

Example

Converting TravelDocs and Flex to Oracle versions

This example illustrates how to convert TravelDocs and Flex applications to Oracle versions. For test purpose, the following server names are used. You must replace the server names, ports, etc. as applicable to your test environment.

- MS Active Directory Server = WIN2K8R2A
- Datacap Server (and the location of the applications) = WIN2K8R2B

- Datacap Web Server = WIN2K8R2C
- Datacap Dev station + Oracle = WIN2K8R2D

On WIN2K8R2D (Oracle)

1. In Oracle server, create a unique database user for each application, so that a schema for that user is created. Log on to Oracle as one of the users created in this step.

Note:

- The users created for test purpose are TravelDocsORA and FlexORA
 - Optionally, you can specify a new tablespace for each user/schema. In this test, the users were created in the system tablespace by default.
 - The users can be in the same tablespace or different tablespaces.
2. Navigate to the location ... \Datacap\support\DBScript\ and execute the following Oracle scripts against each database mentioned above:
 - Oracle_Adm_base.sql
 - Oracle_Eng_base.sql
 - Oracle_FP_base.sql

Repeat the steps 1 and 2 for each application. Optionally, you can create a separate user for each of Oracle_Adm, Oracle_Eng and Oracle_FP. For example, FlexORAadmin, FlexORAengine, and FlexORAfingerprint.

On WIN2K8R2D (DEV Station)

1. Using command line, run the DAppCopy tool as described below to copy the TravelDocs app to TravelDocsORA app:

```
DAppCopy -from -an TravelDocs -af ||WIN2K8R2B\Datacap\TravelDocs -adb
"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=||WIN2K8R2B\Datacap\TravelDocs\TravelDocsAdm.mdb;" -edb
"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=||WIN2K8R2B\Datacap\TravelDocs\TravelDocsEng.mdb;" -fdb
"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=||WIN2K8R2B\Datacap\TravelDocs\TravelDocsFingerprint.mdb;" -how online -to -af
||WIN2K8R2B\Datacap\TravelDocsORA -adb "Provider=OraOLEDB.Oracle.1;Data
Source=WIN2K8R2D;User ID=TravelDocsORA;Password=FileNet123;" -edb
"Provider=OraOLEDB.Oracle.1;Data Source=WIN2K8R2D;User
ID=TravelDocsORA;Password=FileNet123;" -fdb "Provider=OraOLEDB.Oracle.1;Data
Source=WIN2K8R2D;User ID=TravelDocsORA;Password=FileNet123;" -how onlineDAppCopy -from -an
TravelDocs -af ||WIN2K8R2B\Datacap\TravelDocs
```

2. Using command line, run the DAppCopy tool as described below to copy Flex app to FlexORA app:

```
DAppCopy -from -an Flex -af ||WIN2K8R2B\Datacap\Flex -adb "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=||WIN2K8R2B\Datacap\Flex\FlexAdm.mdb;" -edb "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=||WIN2K8R2B\Datacap\Flex\FlexEng.mdb;" -fdb "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=||WIN2K8R2B\Datacap\Flex\FlexFingerprint.mdb;" -how online -to -af
||WIN2K8R2B\Datacap\FlexORA -adb "Provider=OraOLEDB.Oracle.1;Data Source=WIN2K8R2D;User
ID=FlexORA;Password=FileNet123;" -edb "Provider=OraOLEDB.Oracle.1;Data Source=WIN2K8R2D;User
ID=FlexORA;Password=FileNet123;" -fdb "Provider=OraOLEDB.Oracle.1;Data Source=WIN2K8R2D;User
ID=FlexORA;Password=FileNet123;" -how onlineDAppCopy -from -an Flex -af
||WIN2K8R2B\Datacap\Flex
```

On WIN2K8R2B

1. Update the datacap.xml to add TravelDocsORA.
2. Navigate to ... \Datacap\TravelDocsORA folder and rename the Port.app to TravelDocsORA.app.

3. Create export and batches folders in the ...\\Datacap\\TravelDocsORA folder.
4. Install the Oracle 32-bit client.
5. Restart the Datacap Server service.

Repeat these five steps for FlexORA.

On WIN2K8R2D (Oracle)

1. In the TravelDocsORA.admininfo table, update the *db_apptitle* column from TravelDocs to TravelDocsORA. Update the *db_index* column to a different value. (For test purpose, the value is increased by 1).
2. In the FlexORA.admininfo table, update the *db_apptitle* column from Flex to FlexORA. Update the *db_index* column to a different value. (For test purpose, the value is increased by 1)

On WIN2K8R2D (DEV Station)

1. Open Datacap Application Manager and verify/test the connection strings.
2. Test the basic connections to both applications in DStudio and DCDesktop.

On WIN2K8R2C (TMWEB.NET)

1. Edit the ...\\Datacap\\tmweb.net\\apps.ini file to copy the TravelDocs section and copy it as described below.

...from...

[TravelDocs]

Debug=0

Oracle=0

... to...

[TravelDocsORA]

Debug=0

Oracle=1

2. Repeat this step for FlexORA.
Oracle= can be **0** (MSAccess or MSSQL) or **1** (either Oracle or DB2). Change it to **1**.
Note: If the TravelDocs or the Flex sections do not exist, you must manually create them.
3. Restart the IIS web server.
4. Test the tmweb.net for both TravelDocsORA and FlexORA.

Moving the application files

You can use the Datacap Application wizard to copy a Microsoft Access based application and move it to a folder on another computer.

About this task

The following steps are not required to use a DB2®, Microsoft SQL Server, or Oracle database. These steps describe how to copy an application, move it to another environment, and update the appropriate paths. For example, when you want to move an application between Test and Production systems and update the paths to the fingerprints. You can also rename the application when you move it.

You use the Datacap Application Copy Tool copy and move the application and its databases at the same time.

Procedure

To move the application files:

1. Open Start > All Programs > IBM Datacap Developer Tools > Datacap Studio.
2. Click Datacap application wizard.
3. At the Overview window, click Next.
4. Select Copy an existing RRS application and click Next.
5. Select the name of the application that you want to copy. For example, select TravelDocs.
6. In the Root folder on target system field, click [...] and select the database server where you want to copy the application. For example, select C:\Datacap.
7. In the Datacap Web folder field, click [...] and select the folder on the Datacap Client where the application is installed. For example, select C:\Datacap\tmweb.net.
8. Select Rename Copy and type the name of the application in the New Name field.
9. Click Next to collect the information to use to copy the application.
10. Click Finish to copy the application to the target database server.
11. Set the Oracle setting in the apps.ini.
 - a. Go to the Datacap\tmweb.net\apps.ini and copy the *App_name* section.
 - b. Rename the *App_name* section to match your new application name.
 - c. Locate the line *Oracle* in the *App_name* section.
 - d. Change this field to read *Oracle=1* for DB2 or Oracle. For SQL Server, use *Oracle=0*.
 - e. Locate the *DateTimeSeparator* line and change it to read *DateTimeSeparator='*.
 - f. Save your changes.

Parent topic: [Configuring Datacap databases](#)

Verifying the database connection

After you configure the application for the database, verify the connection to the DB2®, Microsoft SQL Server or Oracle database works properly.

Procedure

To verify the database connection:

1. In the Start menu click IBM Datacap Developer Tools > Datacap Studio. If you can log in and run the application by using the Test tab, the Administration and Engine databases are working properly.
2. Click the Zones tab to verify that the Fingerprint database is working.
3. In the Start menu click IBM Datacap Services > Datacap Application Manager. Press Test connection in the Connection string dialog to see if the databases are working properly.

Parent topic: [Configuring Datacap databases](#)

Advanced database settings for Datacap Server

Datacap can use custom connection strings from which it can target any database type that has a supported database provider.

Connection strings

The supported database providers for which you can use these connection strings are Microsoft Access, DB2®, Microsoft SQL Server, and Oracle. The old Datacap format connection strings are still supported to provide compatibility with an earlier version to your existing database connections.

The following examples describe OLEDB connection strings for each of the supported database providers:

DB2 using Standard Authentication

```
"Provider=IBMDADB2;Data Source= database alias; UID=*****; PWD=*****;  
CurrentSchema=db2admin;"
```

DB2 using Windows Authentication

N/A

Microsoft Access using Standard or Windows Authentication

```
"Provider=microsoft.jet.oledb.4.0;data source=C:\Datacap\MyApp\MyAppadm.mdb;  
persist security info=false;"
```

SQL Server using Standard Authentication

```
"Provider=sqloledb;data source=myServerAddress;Initial Catalog= myDataBase;  
User Id=myUsername;Password=myPassword;"
```

SQL Server using Windows Authentication

```
"Provider=sqloledb;data source=myServerAddress;Initial Catalog= myDataBase;  
Integrated Security=SSPI;"
```

Oracle using Standard Authentication

```
Provider=OraOLEDB.Oracle;Data Source=MyOracleDB;User Id=myUsername;  
Password=myPassword;
```

Oracle using Windows Authentication

```
Provider=OraOLEDB.Oracle;Data Source=MyOracleDB;OSAuthent=1;
```

Registry settings

Special cases in which database might act differently than usual are controlled by a set of settings in the registry. Datacap uses these settings when database providers must use alternative logic implementations. Otherwise, Datacap uses the default behavior. There are special settings for different database behavior that is not default.

The following settings are in the InterThread section of TMS registry profile at HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Datacap\TMS\InterThread.

TO_DATE

Formats the date and time field values in TO_DATE format in SQL statements for the providers. This example contains the default TO_DATE setting for Oracle.

```
ORACLE;ODBCORACLE;MSORACLE;ODBCMSORACLE;OraOLEDB.Oracle;MSDAORA;IBMDADB2;DB2OLE  
DB
```

ROWNUM

Use this ROWNUM syntax instead of SELECT TOP in SQL statements for these providers. This example contains the default ROWNUM setting for Oracle.

```
ORACLE;ODBCORACLE;MSORACLE;ODBCMSORACLE;OraOLEDB.Oracle;MSDAORA;IBMDADB2;DB2OLEDB
```

#DATETIME#

Use date and time field values that are wrapped in a # sign-in SQL statements for these providers. Some of the databases want the date and time field values in # signs. This example contains the default #DATETIME# setting for MS Access.

```
MSACCESS;ODBCMSACCESS;Microsoft.Jet.OLEDB.4.0;Microsoft.ACE.OLEDB.12.0
```

"FIELD NAME"

Use field names wrapped in quotes in SQL statements for these providers. Some database types need quoted fields in SQL statements. This example contains the default "FIELD NAME" setting for Oracle.

```
ORACLE;ODBCORACLE;MSORACLE;ODBCMSORACLE;OraOLEDB.Oracle;MSDAORA;IBMDADB2;DB2OLEDB
```

"AUDIT"

Use audit table references wrapped in quotes in SQL statements for these providers. In some databases Audit is a system/internal/key word, so you must wrap Audit in quotes to differentiate these databases. This example contains the default "AUDIT" setting for Oracle.

```
ORACLE;ODBCORACLE;MSORACLE;ODBCMSORACLE;OraOLEDB.Oracle;MSDAORA
```

MOVE_BY_ONE

When you must move the cursor on a recordset more than one row, you do it by moving by one row at the time in a loop. Some providers might have issues trying to move several rows at once, so you must do this loop procedure for them. This example contains the default MOVE_BY_ONE setting for Oracle.

```
ORACLE;ODBCORACLE;MSORACLE;ODBCMSORACLE;OraOLEDB.Oracle;MSDAORA
```

SELECT_COUNT

Use the SQL syntax for SELECT COUNT instead of using ADO API GetRecordCount because GetRecordCount does not work for some databases. This example contains the default SELECT_COUNT setting for Oracle.

```
ORACLE;ODBCORACLE;MSORACLE;ODBCMSORACLE;OraOLEDB.Oracle;MSDAORA
```

RECORD_FORWARD_ONLY

Open ADO recordsets by using the forward only setting. The default setting is static bidirectional, but some databases have issues with static bidirectional recordsets. This example contains the default RECORD_FORWARD_ONLY setting for Oracle.

```
ORACLE;ODBCORACLE;MSORACLE;ODBCMSORACLE;OraOLEDB.Oracle;MSDAORA
```

Application settings - {default} option

The Datacap Application Manager is updated so that you can select an option as {default} for some application on the Application settings page.

- The Authenticator list shows the ADLDS option in the list. Selecting ADLDS must take effect on any setting on the Task master server side.
- Added {default} option for following fields in application manager:
 - Queue by
 - Role-based batch filtering
 - Authenticator

- If the {default} option is set for these fields, then application picks up the setting for those fields from the Taskmaster server manager.
- For "Role-based batch filtering" if the {default} option is set in application manager, then it picks up the setting from the registry.

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Datacap\TMS\InterThread -> Role Mode

Where, Role Mode represents the following values:

- 0 = None
- 1 = Exclusive
- 2 = Additive
- 3 = Inclusive



Upgrading

Migrating a Datacap environment to a new release requires that you upgrade the component software and migrate your deployed applications. Depending on the currently installed version of Datacap, upgrading the software may first require removing the previous version before installing the new release. The migration steps can include revising existing functionality, adding new features, or changing baseline user interfaces.

Before you begin

For hardware and software requirements, see [System Requirements: Hardware and Software Requirements for IBM® Datacap, Version 9.0.1](#).

About this task

Upgrades from the following IBM Datacap versions are supported:

- IBM Datacap V8.0.1
- IBM Datacap V8.1.0
- IBM Datacap V9.0 or later version

Procedure

To upgrade to the latest Datacap software version:

1. Create the test environment. See [Creating a test environment](#).
2. Create the production environment. See [Creating a production environment](#).

Parent topic: [Installing](#)



Creating a test environment

Install Datacap in a test environment to deploy, configure, and test any modifications.

Procedure

To create a test environment:

1. In a non-production development environment, do the following actions based on your previous Datacap version:

Option	Description
V9.0	a. Back up your Datacap application folders. b. If you modified any of the standard Datacap applications, make a backup of those folders also. c. On each computer that you are upgrading, uninstall Datacap V9.0. (Uninstalling Datacap V9.0 preserves the files that you added or modified). Tip: If you already installed Datacap V9.0.1 without first uninstalling Datacap V9.0, see Repairing a Datacap 9.0.1 installation .
All others	Uninstalling your previous version of Datacap is not necessary.

- In a non-production development environment, install the latest version of Datacap. You might use new interfaces that are available with the latest Datacap version. For example, the Datacap thick client in version 8.0.1 is replaced by Datacap Desktop for the end users, and by Datacap Web Client and IBM® Content Navigator for the administrators. For installation details, see [Datacap installation and configuration in a client/server environment](#).
- Import encryption keys as necessary. You must have FIPS encryption as of version 8.1.0 for password protection. For more information, see [Importing encryption keys to Datacap computers](#).
- Based on your previous Datacap version as shown in the following table, migrate and copy your customizations from your previous Datacap environment to this test environment:

Option	Description
V8.0.1	See Upgrading Datacap applications from Taskmaster 8.0.1 .
V8.1	See Upgrading Datacap applications from Taskmaster 8.1 .
V9.0	See Upgrading Datacap applications from 9.0 .

- Consult the What's New topics to determine whether any existing rules or actions were eliminated or deprecated in the new release. For more information, see [What's new](#). Alternatively, open Datacap Studio and determine whether any existing rules or actions were eliminated or deprecated in the new release. Deprecated actions are designated as Private and they cannot run in the Test tab in Datacap Studio. But they can run in your updated applications. In a subsequent release, remove these deprecated actions from the application.
- Modify your customized settings as needed. Based on your previous Datacap version as shown in the following table, see the indicated topic:

Option	Description
V9.0 feature pack 1 and 2	See Setting Datacap Navigator default layouts in Datacap 9 and Feature Packs 1 and 2 .
Any version before V9.0 feature pack 2	See Migrating Datacap Navigator custom panels to Datacap 9.0 Feature Pack 2 or later .

- Update the code and customized user interfaces that were not fully migrated to maintain your current level of function. For more information, see [Customized panel conversion to Datacap Desktop](#).
- Apply any new features in the updated software version that you want to implement.
- Test your updated applications and configuration in the development environment against your current production system. You run this parallel testing to validate features, performance, connectivity, and volume throughput. For example, ensure that program communication works as intended between servers and clients, web or otherwise.



Upgrading Datacap applications from Taskmaster 8.0.1

If you want to use applications that were created in Taskmaster 8.0.1, use the Application Wizard to convert these applications to work with the latest Datacap software.

Procedure

To migrate an application to the target computer with the new Datacap software version:

1. Copy the entire folder of the application that you want to convert to the \Datacap folder on the target computer.
2. On the target computer, open `datacap.xml` and add the name of the application that you want to update. For example, if the application name is *MyApp*, copy the *MyApp* folder to \Datacap.
3. If you are using DB2®, Oracle, or Microsoft SQL Server, create a new database by using the creation scripts for each database type (administration, engine, fingerprint).
Attention: The creation scripts are in the `C:\Datacap\support` folder.
4. If you are using a Microsoft Access database, complete these steps:
 - a. On the target computer, copy `FormTemplateAdm.mdb`, `FormTemplateEng.mdb`, and `FormTemplateFingerprint.mdb` from `C:\Datacap\Templates\FormTemplate` to a temporary folder. The databases that you are copying are blank, but they contain the target database table formats. If the existing batches of documents are not needed in the converted application, you can use a blank Engine database. If you need the batches of documents, complete the batches before you convert the application.
 - b. Rename the three databases Adm, Eng, Fingerprint in the temporary folder to the name of the application that you are converting. For example, if the application name is *MyApp*, rename the three databases to `MyAppAdm.mdb`, `MyAppEng.mdb`, and `MyAppFingerprint.mdb`.
 - c. If you customized any of the Taskmaster 8.0.1 databases, you must add any custom columns and new columns from the Taskmaster 8.0.1 databases to the target databases.
Important: Ensure that the column names that you are adding to the target databases match the column names in the 8.0.1 databases.
5. Start Datacap Studio on your workstation by clicking `Start > IBM Datacap Developer Tools > Datacap Studio`.
6. In the Datacap Studio startup dialog that prompts you to select an application, click Close.
7. In Datacap Studio, click the Datacap application wizard icon.
8. In the Application Wizard Overview window, click Next.
9. Select Convert an 8.0.1 application to 9.0 format.
10. In the Source Application window, select the application that you want to convert.
11. In the 9.0 Target Databases dialog, enter the database information that your updated application must use:
 - o If you are using a Microsoft Access database, browse to the temporary folder where you copied the databases and select the three databases.
 - o If you are using a DB2 database, enter the service name of the DB2 source database. If the user ID that created the database tables is different from the user ID in the connection string, you must also specify the user ID that created the database tables as the schema. To specify the correct schema, use the user ID in the connection string. Use `CurrentSchema=user ID that created the tables`.

Important: You must enter information for each of the databases: Administration, Engine, and Fingerprint. Click the corresponding tab to enter database information.

When you upgrade a custom Datacap application from Taskmaster 8.0.1 to 9.0 or later, if the Datacap application wizard fails to upgrade the application, complete the following steps to resolve this issue:

- a. Open the application app file. The app file is at `C:\Datacap\application\application.app`
- b. Copy the following Taskmaster 8.0.1 plain connection string in the `application.app` file:

```
PROVIDER=MSACCESS;DSN=C:\Datacap\application\applicationAdm.mdb;
```

For example, copy the

```
PROVIDER=MSACCESS;DSN=C:\Datacap\application\applicationAdm.mdb;
```

string in the attribute that has attribute value as "tadmin."

c. Similarly, update the following attribute values by adding corresponding plain strings:

- tmengine
- fingerprintconn

d. Save and close the app file.

e. Open the Datacap application wizard.

f. Select Convert a Taskmaster 8.0.1 application to Datacap 9.0 or later, and click Next.

g. Select the custom application, and select Next.

h. Browse to the Administration, Engine, and Fingerprint databases connection strings, and then click Next. The Application wizard completes the migration.

When the plain text connection strings are set manually while running the wizard, the string in the app file gets modified as:

```
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=[path/]\applicationEng.mdb;
```

12. Click Next and then Finish. The Application wizard displays a message to indicate a successful update, and whether you must review the Application wizard log for any errors or warnings. The log file is in C:\Datacap\application name\appwiz_convert.log.

Important: During the conversion of your application, Datacap makes the following modifications.

- Tasks that were previously mapped to use prelayout.aspx are now mapped to use verifine.aspx.
- Tasks that previously used project files are configured through the task setup window. You access task setup by selecting the task on the Workflow page of the Administrator tab in the Datacap Web Client, and clicking Setup.
- The databases in the temporary folder are merged with the databases in the application folder. The databases in the temporary folder now contain the Taskmaster 8.0.1 application data in the target tables.

13. If you are using a Microsoft Access database, copy the three databases in the temporary folder to the application folder and select Yes, overwrite existing files at the prompt. For example, the application folder might be C:\Datacap\MyApp.

14. Select Start > IBM Datacap Services > Datacap Application Manager to start Datacap Application Manager and confirm that the three databases are pointing to the correct data source.

15. If necessary, change the path to the correct data source and restart the Datacap Server Manager on the Service tab of the Datacap Application Manager.

16. Restart Datacap Studio, log on to the application that you converted, and confirm that you can connect to the Administration and Engine databases.

17. Load the application in Datacap Studio and correct any action or rule errors:

- a. Click Start > IBM Datacap Developer Tools > Datacap Studio.
- b. Select the application that you want to update, click Next, and log on.
- c. Correct any errors that are displayed such as warning messages or action errors that are highlighted in red in the Ruleset window.

18. Update settings for all tasks that were configured for DotScan, DotEdit, or the multiple program option in Taskmaster 8.0.1. Using Datacap Web Client or Datacap Navigator administration view, go to Workflows, select the task, and click Settings. Verify or modify settings as required, and then save.

19. Optional: Port Datacap Web Client tasks in the application to Datacap Navigator. You must configure task settings for Datacap Web Client tasks to work in Datacap Navigator. Some Datacap Web Client tasks port directly to Datacap Navigator. For example, *rScan* corresponds to Scan, and *Verifine* corresponds to Verify. However, other Datacap Web Client tasks (such as *ProtoID*, *ImageEnter*, and *averify*) do not have direct equivalents in Datacap Navigator. If you need to port a customized task from Taskmaster Web to Datacap Navigator, contact IBM® Professional Services for assistance.

To port simple Datacap Web Client tasks to Datacap Navigator:

- a. Ensure that the Datacap Navigator plug-in is deployed and a repository is created for the application. For instructions, see [Installing and configuring Datacap Navigator](#).
- b. Open the Datacap Navigator administration view by using the following URL:

```
http://server:port/navigator/?desktop=dcadmin
```

- c. In the left pane, click Workflows. Create a workflow with jobs and tasks that correspond to your tasks in Datacap Web Client. Select appropriate program options for Datacap Navigator tasks as follows:

Datacap Navigator task	Program option
Scan	scan.js
Upload	upload.js
Verify	verify.js
Fixup	classify.js

- d. In the left pane, click Shortcuts. Create shortcuts for Datacap Navigator tasks.



Upgrading Datacap applications from Taskmaster 8.1

You can upgrade Taskmaster 8.1 applications by copying them to a server with the latest Datacap software version. Taskmaster 8.1 applications with Datacap Web Client (formerly known as Taskmaster Web Client) and Rulerunner tasks only will work without further configuration. Taskmaster 8.1 applications with DotScan and DotEdit tasks require adjustments to task settings.

Procedure

To migrate an application from Taskmaster 8.1 to the target computer with the new Datacap software version:

1. Copy the application that you want to migrate to the target computer by using the Datacap Application Copy Tool:
 - a. Ensure that your new Datacap system has access to the computer on which the Taskmaster 8.1 application and its databases reside.
 - b. Ensure that the processing of all batches is complete before you migrate the application.
 - c. Create new, empty databases for administration, engine, and fingerprint.

If you are using DB2®, Oracle, or Microsoft SQL Server, create new target databases by using the creation scripts for each database type (administration, engine, fingerprint). The creation scripts are in the C:\Datacap\support\DBScript directory. If custom columns were added to the tmbatch table in the Taskmaster 8.1 database, add them to the target creation scripts.

If you are using Microsoft Access, see [Updating Microsoft Access database schemas from Taskmaster 8.1](#).

- d. Click Start > IBM Datacap Services > Datacap Application Manager. On the Service tab in the Path to the application management file field, specify the path to the datacap.xml file in the Taskmaster 8.1 environment.
- e. Click Start > IBM Datacap Developer Tools > Datacap Application Copy Tool and specify the following settings:

- In the Copy from area on the left, select the Taskmaster 8.1 application to copy. The path to the Application files and the connection strings for the Administration and Fingerprint databases are populated.
- In the Copy to area on the right, select Copy to portable data files, leave the Document Hierarchy and Rules options selected, set the Application files path to a folder that does not exist (the tool will create the folder), and leave the database settings empty.

Then, click OK to run the Datacap Application Copy Tool, and click OK on the confirmation message that says "Do you want to replace or update the '*application_files_path*' application?" The application folder will be created. In the Copy to area, if you specified a folder that already exists, the copy fails and you must run the tool again.

f. Start Datacap Application Manager. On the Service tab in the Path to the application management file field, specify the path to the datacap.xml file in the version 9.0 or later environment.

g. In the Datacap Application Copy Tool, specify the following settings:

- In the Copy from area on the left, select Copy from portable data files and specify the temporary application folder in the Application files folder.
- In the Copy to area on the right, select Add new application and set the path to a folder that does not exist (the tool will create this folder), leave the Document Hierarchy and Rules options selected, and specify connection strings for the administration and fingerprint databases. Leave the engine database setting empty unless you need to transfer existing batches to the migrated application. In this case, you should select the Clear Engine database option and specify the connection string for the engine database. If you are using Microsoft Access, you should specify the renamed FormTemplate databases in the temporary folder.

Then, click OK to run the Datacap Application Copy Tool to copy the application files and database contents to the folder and databases of the migrated application.

2. Configure environment settings by using Datacap Application Manager:

- a. Click Start > IBM Datacap Services > Datacap Application Manager.
- b. In the left pane, select the application that you want to update.
- c. In the right pane, update settings on the Main tab and ensure that any settings on the Custom values tab are correct for the Datacap 9.0 target environment.

If you are using Microsoft Access, the application configuration now points to the renamed and updated FormTemplate databases in the temporary folder. You should move these databases to the folder of the migrated application and update the connection strings in Datacap Application Manager.

If you are using repositories or additional databases such as the lookup and export databases, configure the repositories or databases in the migrated environment. Use the same instances, or create new instances to avoid interfering with the Taskmaster 8.1 environment. Then, update settings in Datacap Application Manager.

3. Load the application in Datacap Studio and correct any action or rule errors:

- a. Click Start > IBM Datacap Developer Tools > Datacap Studio.
- b. Select the application that you want to update, click Next, and log on.
- c. Correct any errors that are displayed such as warning messages or action errors that are highlighted in red in the Ruleset window.

4. Update settings for all tasks that were configured for DotScan, DotEdit, or the multiple program option in Datacap Taskmaster 8.1. Using Datacap Web Client or Datacap Navigator administration view, go to Workflows, select the task, and click Settings. Verify or modify settings as required, then save.

5. Optional: Port Datacap Web Client tasks in the application to Datacap Navigator. You must configure task settings for Datacap Web Client tasks to work in Datacap Navigator. Some Datacap Web Client tasks port directly to Datacap Navigator. For example, rScan corresponds to Scan, and Verifine corresponds to Verify. However, other Datacap Web Client tasks (such as ProtoID, ImageEnter, and averify) do not have

direct equivalents in Datacap Navigator. If you need to port a customized task from Taskmaster Web to Datacap Navigator, contact IBM® Professional Services for assistance.

To port simple Datacap Web Client tasks to Datacap Navigator:

- a. Ensure that the Datacap Navigator plug-in is deployed and a repository is created for the application. For instructions, see [Installing and configuring Datacap Navigator](#)
- b. Open the Datacap Navigator administration view by using the following URL:

```
http://server:port/navigator/?desktop=dcadmin
```

- c. In the left pane, click Workflows. In the right pane, select a workflow and click Edit. Then, create a job and tasks that correspond to your tasks in Datacap Web Client. Select appropriate program options for Datacap Navigator tasks as follows:

Datacap Navigator task	Program option
Scan	Scan.js
Upload	Upload.js
Verify	Multiple
Fixup	Multiple

- d. On the Task > Advanced tab, scroll down to the Datacap Navigator area and specify Web Program options for Verify and Fixup tasks as follows:

Datacap Navigator task	Web Program option on the Advanced tab
Verify	Verify.js
Fixup	Classify.js

- e. In the left pane, click Shortcuts. Create shortcuts for Datacap Navigator tasks.



Updating Microsoft Access database schemas from Taskmaster 8.1

The Microsoft Access database schemas must be updated.

Procedure

To update Microsoft Access database schemas:

1. On the target computer, ensure that the FormTemplate application was not used to create batches or customized. Then, copy FormTemplateAdm.mdb, FormTemplateEng.mdb, and FormTemplateFingerprint.mdb from C:\Datacap\Templates\FormTemplate to a temporary folder.
2. Rename the three databases Adm, Eng, Fingerprint in the temporary directory to the name of the application that you are converting. For example, if the application name is *MyApp*, rename the three databases to MyAppAdm.mdb, MyAppEng.mdb, and MyAppFingerprint.mdb.
3. If you customized any of the Taskmaster 8.1 databases, you must add any custom columns to the target databases.



Upgrading Datacap applications from 9.0

To upgrade a Datacap application from 9.0.0, you must update the application database schemas. If your application contains compiled rulesets, you must update any rulesets that were updated in the latest Datacap version. You can add accuracy statistics to an existing application. Datacap Navigator settings for certain tasks must be updated. If your application contains customized Desktop panels, you must rebuild them by using the updated project on DeveloperWorks.

Procedure

To upgrade Datacap applications from 9.0:

1. If you modified any of the standard applications in Datacap 9.0 such as the Accounts Payable application, you might want to preserve those changes. To do so, restore those application folders from the backups that you made in step 1.
2. If your application uses DB2, SQL Server, or Oracle databases, update the database schemas "in place" (without removing an existing older version):

DB2	<ol style="list-style-type: none"> a. Create full backups of the Admin, Engine, and Fingerprint database tables. b. Start IBM Data Studio. c. Connect to the application's databases. d. Execute the following scripts in the \Datacap\support\DBScript\ folder: <ul style="list-style-type: none"> ▪ DB2_Adm_Upd901.sql ▪ DB2_Eng_Upd901.sql ▪ DB2_FP_Upd901.sql
SQL Server	<ol style="list-style-type: none"> a. Create full backups of the Admin, Engine, and Fingerprint database tables. b. Start SQL Server Management Studio. c. Connect to the application's databases. d. Execute the following scripts in the \Datacap\support\DBScript\ folder: <ul style="list-style-type: none"> ▪ SQL_Adm_Upd901.sql ▪ SQL_Eng_Upd901.sql ▪ SQL_FP_Upd901.sql
Oracle	<ol style="list-style-type: none"> a. Create full backups of the Admin, Engine, and Fingerprint database tables. b. Start SQL Plus or SQL Developer. c. Connect to the application's databases. d. Execute the following scripts in the \Datacap\support\DBScript\ folder: <ul style="list-style-type: none"> ▪ Oracle_Adm_Upd901.sql ▪ Oracle_Eng_Upd901.sql ▪ Oracle_FP_Upd901.sql

3. If your application uses Microsoft Access databases, see [Updating Microsoft Access database schemas from Datacap 9.0](#).
4. If your application contains any of the following rulesets, manually copy the updated versions from the \Datacap\rrs folder to your application's Rules folder:
 Important: A bug-fix or an enhancement that has been added to a ruleset DLL will not be available in an application until the updated ruleset DLL is manually copied from the RRS folder to the application Rules folder. For example, your application's Rules folder might be \Datacap\TravelDocs\dco_TravelDocs\Rules\.

Ruleset	File to copy
---------	--------------

Ruleset	File to copy
Populate Fields Using Keywords	Populate_Fields_Using_Keywords.Rul.dll
ExportToBoxRul	ExportToBoxRul.dll
ImportFromBoxRul	ImportFromBoxRul.dll
RecognizePagesAndFieldsRul	RecognizePagesAndFieldsRul.dll

5. You might want to add statistics collection to an existing application from Datacap 9.0. Adding statistics collections is necessary to support the Dashboard feature for Datacap Navigator, count processed checks, and for other purposes. To add statistics collection, do the following steps:
 - a. Copy the files Profile Statistics.rule and Export Statistics.rule to \Datacap\RRS.
 - b. Use FastDoc to drag the Profile Statistics and Export Statistics rulesets to your application. For example, manually copy
 \Datacap\Templates\FormTemplate\dco_FormTemplate\rules\ProfileStatistics.rule to
 \Datacap\RRS, and then use FastDoc to copy that ruleset to your application.
6. Load the application in Datacap Studio and correct any action or rule errors:
 - a. Click Start > IBM Datacap Developer Tools > Datacap Studio.
 - b. Select the application that you want to update, click Next, and log on to the application.
 - c. Correct any errors that are displayed such as warning messages or action errors that are highlighted in red in the Ruleset window.
7. To use new Datacap Navigator features, re-enter task settings for the following task programs:
 - o Scan.js
 - o Classify.js
 - o Verify.js
 - o Multiple

The following steps are required for each task that uses these Datacap Navigator task settings. If a task is used in more than one job, you need to re-enter task settings for one job only. To re-enter task settings:

- a. Open the Datacap Administration View. In a browser, enter a URL with the following format:

```
http://host_name:port_number/context_root/?desktop=dcadmin.
```

By default, the context root is navigator.

- b. In the left pane, click Workflows.
 - c. In the right pane, select a workflow and click Edit.
 - d. Click the Jobs tab, select a job, and click Edit.
 - e. Click the Tasks tab, select a task, and click Edit.
 - f. On the Task page, click the Advanced tab. Make a note of all task settings.
 - g. Click the General tab and change the Program to Rulerunner. For example, if Program is set to scan.js, select Rulerunner from the drop-down list.
 - h. Click Save to reset all task settings.
 - i. Change Program back to the original setting. For example, change it back to scan.js.
 - j. Click the Advanced tab.
 - k. Re-enter the task settings that you noted in step f.
 - l. Click Save.
8. If your application contains customized screen panels for Datacap Desktop, you must rebuild them with the updated SDK for 9.0.1. See [Rebuilding customized screen panels for Datacap Desktop](#).



Rebuilding customized screen panels for Datacap Desktop

If your application contains customized screen panels for Datacap Desktop, you must rebuild them with the updated SDK. This procedure requires Microsoft Visual Studio 2013.

Procedure

To rebuild customized screen panels:

1. Download the Datacap V9.0.1 Custom Panel project from IBM DeveloperWorks. See [IBM® Datacap 9.0 and Datacap 9.0.1 DDK Datacap Desktop Custom Panels](#).
2. Start Microsoft Visual Studio 2013.
3. Open the DCDesktopPanels solution.
4. Add the custom source files to your project that were used by your Datacap V9.0 custom panels:
 - a. Copy the files to your project folder.
 - b. Select the panel project.
 - c. Choose menu item Project > Add Existing Item and select the files that you copied.
5. Build the solution. If necessary, adjust your code to be compatible with the new Datacap software version.
6. Deploy your custom DLL to the application and test.



Repairing a Datacap 9.0.1 installation

This procedure is required only if you installed Datacap 9.0.1 without previously uninstalling an existing 9.0 system on that computer.

Procedure

To repair a Datacap 9.0.1 installation:

1. Remove the leftover 9.0 components:
 - a. From Control Panel > Programs > Programs and Features, uninstall any of the following items if they exist:
 - IBM® Datacap 9.0
 - IBM Datacap 9.0.0.1
 - IBM Datacap 9.0.0.2
 - IBM Datacap 9.0.0.3

Important: Do not uninstall IBM Datacap 9.0.1.
2. Delete any of the following folders and files if they exist. (They are restored as part of the Datacap 9.0.1 installation.)

Folders	<ul style="list-style-type: none">○ \Datacap\dcshared\ICRC○ \Datacap\dcshared\OCRA\Data\ExtendedDictionaries○ \Datacap\Flex○ \Datacap\Templates○ \Datacap\TravelDocs
Files	<ul style="list-style-type: none">○ \Datacap\dcshared\Accusoft.BarcodeXpress9.ActiveX.dll○ \Datacap\dcshared\NET\Accusoft.ScanFixXpress7.Net.dll

3. Install Datacap 9.0.1 in the same location that you uninstalled Datacap 9.0:
 - a. With Administrator privileges, run the Datacap 9.0.1 installer from the command line as shown here:

```
setup.exe /V"REINSTALLMODE=amus REINSTALL=ALL"
```

- b. With Administrator privileges, run the following command from the command line to reregister the Datacap Taskmaster Server service:

```
\Datacap\Taskmaster\TMS.exe /Service
```



Updating Microsoft Access database schemas from Datacap 9.0

This procedure requires Microsoft Access 2013 or later version.

Procedure

To update the Microsoft Access database schema for the target Datacap system with the latest Datacap version:

1. Replace the 9.0 Engine database file with an empty Engine database for the target system:
Important: This step does not preserve existing batches.
 - a. Back up the Datacap 9.0 Engine database file in your application folder by renaming it or copying it to a backup location.
 - b. Copy the Engine database file from the target system's Form Template application (such as \Datacap\Templates\FormTemplate) and paste it into your application folder.
 - c. Rename the target system's Engine database file that you copied to match the original file name. (The original file name is the name before any renaming done in step 1a.)
 - d. Delete existing batch folders from the application.
2. If your application contains custom columns, use Microsoft Access to add your custom columns to the tmbatch table in the new Engine database.
3. Update the Fingerprint database schema to support the ClassifyLayout actions:
 - a. Start Microsoft Access and open the following file:
\Datacap\Templates\FormTemplate\FormTemplateFingerprint.db (Datacap 9.0.1 schema).
 - b. Start another copy of Microsoft Access and open the application's existing Datacap 9.0 Fingerprint database file.
 - c. Copy the Class, Feature, and Knowledge tables from the target system's schema and paste them into the existing Fingerprint database.
4. Update the Administrative database schema with tables and fields reserved for future use:
 - a. Start Microsoft Access and open the following file:
\Datacap\Templates\FormTemplate\FormTemplateAdm.mdb (the target system's schema).
 - b. Start another copy of Microsoft Access and open the application's existing Datacap 9.0 Administrative database file.
 - c. Copy the views table from the target system's schema and paste it into the existing Administrative database.
 - d. Close the target system's FormTemplateAdm database.
 - e. Open the existing adminfo table in design view and add a Number type field that is named db_keymode. Set the Field Size to Long Integer.
 - f. Open the existing tmgroup table in design view and add the long integer field gr_key.



Upgrading Datacap applications from Datacap 9.1.1

The sequence to upgrade Datacap applications from Datacap 9.1.1 to Datacap 9.1.3 is - a. Take a backup of required files, b. Install Datacap 9.1.3, c. Start Datacap Taskmaster Service, d. Launch Datacap applications.

Procedure

The detailed procedure to upgrade to Datacap 9.1.3 is as follows -

1. During a maintenance window, take a backup of existing Admin, Engine, and Fingerprint database tables, custom applications including any modified out of the box Datacap applications, custom actions, RuleRunner configuration file (RuleRunner.xml), datacap.xml to a temporary location before upgrading.
2. Run the installer to install IBM Datacap 9.1.3. It automatically upgrades Datacap 9.1.1 applications to Datacap 9.1.3.
3. Confirm that the user credentials are valid for running Datacap Taskmaster Service.
4. After the upgrade is successful, launch the Datacap applications. You can start processing batches.

Note:

- You do not need to uninstall Datacap 9.1.1 before upgrading to Datacap 9.1.3.
- Reconfigure Datacap accounts and groups that are based on your authentication system.
- Reconfigure connection strings to applications in Datacap Application Manager (where applicable).
- Set up sharing and security permissions for users, Datacap folders, and Datacap applications.
- Previously generated encryption keys in the keystore are preserved and need not be generated again.
- Review Installer.log file that is located at C:\Users\localadm\AppData\Local\Temp to confirm that there are no errors.



Migrating Datacap Navigator custom panels to Datacap 9.0 Feature Pack 2 or later

Starting with IBM® Datacap Version 9.0 Feature Pack 2, Datacap Navigator custom panels are stored in a new location.

About this task

In Datacap Version 9.0 and Feature Pack 1, Datacap Navigator custom panels were stored in the following folder:

```
\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\node_name\  
navigator.ear\navigator.war\dcpanels\application_name\
```

In Feature Pack 2, Datacap Navigator custom panels are stored in the following folder:

```
Datacap_installation_folder\application_name\navigatorpanel\panels
```

If you created custom panels in Datacap Version 9.0 or Feature Pack 1 and want to use them in 9.0 Feature Pack 2 or later, you must move your panels to the new location and create an XML file that names the custom panels.

Procedure

To migrate Datacap Navigator custom panels to Datacap Version 9 Feature Pack 2 or later:

1. Copy the custom panel (JSON file) that you want to migrate from this location:

```
\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\node_name\  
navigator.ear\navigator.war\dcpanels\application_name\  

```

2. Paste the custom panel file into this folder:

```
Datacap_installation_folder\application_name\navigatorpanel\panels
```

3. Remove the *.json file extension from your custom panel file name. For example, if the name of your custom panel file is MyPanel.json, change the file name to MyPanel.
4. Using a text editor, create an XML file that names the custom panel. For example, if the name of your custom panel is MyPanel, create an XML file with the following content:

```
<Panels>  
  <Panel name="MyPanel" />  
</Panels>
```

5. Name the file panels.xml and save it in the following folder, on the same level as the panels folder:

```
Datacap_installation_folder\application_name\navigatorpanel\  

```

Important:

Ensure that the name of your custom panel is specified in the same way in the panels folder, in the panels.xml file, and in the Name field on the Datacap Navigator Administration View > Panel page.



Setting Datacap Navigator default layouts in Datacap 9 and Feature Packs 1 and 2

The Classify, Verify, and Scan pages contain widgets such as the image viewer, start panel, field details, and batch structure. Starting with Datacap 9 Feature Pack 3, you can set the default location of the widgets for specific tasks. To enable this functionality for tasks from Datacap 9 and Feature Packs 1 and 2, edit the task configuration XML file.

Procedure

1. In a text editor, open the *task.set.xml* file that is located in the *Datacap_install\application\dco_application* folder, for example:

```
C:\datacap\TravelDocs\dco_TravelDocs\nverify.set.xml
```

2. Paste the following text at the end of the file, before the final `</S>`:

```
<P n="NCLayout" tip="Defines runtime layout" title="Navigator Layout">  
<V label="Layout" n="Layout" tip="Defines the runtime layout" type="text"></V>  
</P>
```

3. Save and close the *task.set.xml* file.



Upgrading Datacap applications to 9.1.0

You can upgrade the Datacap applications to 9.1.0.

Procedure

To upgrade your Datacap application to 9.1.0, complete the following steps:

1. Ensure to take full backup of the Admin, Engine, and Fingerprint database tables.
2. Ensure that the Admin, Engine, and Fingerprint databases are transferred to SQL Server, Oracle, and DB2.
3. Run the installer to install IBM Datacap 9.1.0.
Note: You do not need to uninstall IBM Datacap 9.0.1 before upgrade.
4. Use any of the following three cases for migration of Datacap 9.0.1 applications to 9.1.0. You might need to update database schemas. Two cases require DB changes and one does not require changes to the DB.
 - o Case A - Upgrade from 9.0.1 to 9.1 does not require DB schema changes.
 - o Case B - If you upgrade from Taskmaster 8.1 or 9.0 to 9.1, you must complete the DB migration steps for 9.0 upgrade to 9.0.1. For more information, see [Updating Microsoft Access database schemas from Datacap V9.0 to V9.0.1](#).
 - o Case C - If you upgrade Datacap applications from Taskmaster 8.0.1 to 9.1, you must complete all the following process:
 - [Migrating Datacap applications from 8.0.1 to 9.0 or later](#)
 - [Updating Microsoft Access database schemas from Datacap V9.0 to V9.0.1](#)Note: A bug-fix or an enhancement that is added to a ruleset DLL is not available in an application until the updated ruleset DLL is manually copied from the RRS folder to the application Rules folder. If your application contains any of the following rulesets, you must manually copy the updated versions of the rulesets to your application's Rules folder: Populate Fields Using Keywords, ExportToBoxRul, ImportFromBoxRul, and RecognizePagesAndFieldsRul.
5. Open the \Datacap\rrs folder, and copy and paste the following files into the Rules folder of your application. For example, your application's Rules folder can be \Datacap\TravelDocs\dco_TravelDocs\Rules\
 - o Populate_Fields_Using_Keywords.Rul.dll
 - o ExportToBoxRul.dll
 - o ImportFromBoxRul.dll
 - o RecognizePagesAndFieldsRul.dll
6. After the upgrade is complete, restart the computer.
7. Rebuild and reinstall your customized Datacap Desktop panels, if any. [The Datacap Desktop panels are for use with IBM Datacap 9.1.](#)

What to do next

After you upgrade Datacap applications to 9.1.0, complete the following steps:

1. Validate that the upgrade to 9.1.0 is successful with no errors.
2. Validate that the Install.log contains no errors or warnings.
3. For each application upgraded to 9.1.0, validate that batch records are intact and have identical details compared with 9.0.1 FP2 using the following clients:
 - o Datacap Desktop
 - o Datacap Studio
 - o Fastdoc
 - o TMWeb
 - o RV2

The Datacap Desktop panels are for use with IBM Datacap 9.1

This project contains a modifiable version of the universal verify panel. DCDesktop can be configured to use the built-in universal panel or use the customized universal panel that is contained in this project. Any of the other customized panels can be displayed as well, as in previous releases.

To use this project, IBM Datacap 9.1 must be installed on the development computer.

Follow the directions in the guide to build the panel project. The project must be built on a computer that has IBM Datacap 9.1 installed. The build might initially fail due to references that are not located by the compiler. To help the compiler find the .NET dependent DLLs, in each of the project files add a "Reference Path" to the DCDesktop folder, such as C:\Datacap\DcDesktop\. If the compiler still cannot find the referenced .NET DLL, then right-click the DLL, and select "Remove", click "References", and select "Add Reference". Then, browse to the DLL and add it as a reference.

If the compiler cannot find a DLL referenced through COM, then delete the COM reference from the project and readd the COM reference. Delete the reference by right-clicking on it and selecting "Remove", then right-click "References", select "Add Reference", and select the COM library from the COM section.

The panel guide also includes steps to place your custom panel into its own DLL instead of placing it into the DotEditPanel.dll.

If you have existing panels which change to files that are provided in the template, it is recommended that you start fresh with the new template and reapply your changes to the template. If you replace files in the template with older versions of the files from an earlier project, your project might not run successfully.



Customized panel conversion to Datacap Desktop

If you used customized Batch Pilot or DotEdit panels in Datacap, you must convert those panels so that they can be used in Datacap Desktop on the latest version of Datacap.

About this task

If you do not have Datacap V8.0.1 installed on a separate computer, you must convert the Batch Pilot panel before you can upgrade.

Procedure

To convert the customized panels:

1. Generate the layout XML file for the panel. See [Generating the layout XML file](#).
2. Open the layout XML file in a text editor. For more information, see [The layout XML file](#).
3. Convert the panel by using Microsoft Visual Studio. See [Creating the Datacap Desktop in Microsoft Visual Studio](#).



Generating the layout XML file

You can convert your customized panels to Datacap Desktop panels. You must generate a layout XML file that contains the panel to convert and the name and location of the converted panel.

Procedure

To generate the layout XML file:

1. Start the client for which you want to convert panels, Batch Pilot, or DotEdit.
2. Click File > Open Project.
3. Select the Verify project of the application and click Open. For example, `\Datacap\Survey\dco_Survey\verify.bpp`.
4. In the Batch View window, expand the application to the page level so that the pages are visible.
5. Right-click the page that you want to convert and choose View Form.
6. Click Form > Run Script.
7. Press Shift+Alt+S to run the script that exports the form information as a layout XML file.
8. Select the target location and specify the file name.
9. Click Open to generate the XML file.



The layout XML file

After you generate the layout XML file, you can open the file in a text editor like Notepad. You can review the control types that are listed in the file.

The layout XML file contains information about each of the standard Datacap controls that are used. The file includes the control type, control name, position, and other attributes. The script exports only the following control types.

- dcimage
- dcredit
- combobox
- label

If your form includes other control types, you must manually add them by using Microsoft Visual Studio.



Creating the Datacap Desktop in Microsoft Visual Studio

After you generate and review the layout XML file, open the layout XML file in Microsoft Visual Studio. Then, you can convert the customized panel to the Datacap Desktop panel.

Procedure

To create the Datacap Desktop form in Microsoft Visual Studio:

1. In Visual Studio, open the Datacap Desktop project and press Ctrl+F5.
2. From the Select User Control list, select the DotEditPanels.dotMaster.
3. In the DCO Setup field, click Browse and select the application DCO XML file. For example, `C:\Datacap\MyApp\dco_MyApp\MyApp.xml`.
4. In the Layout XML field, click Browse and select the XML file that you generated.
5. In the Page Type field, select the page for which to create the custom panel.
6. In the New name field, enter the name for the new C# class.
7. Click Create. Datacap Desktop displays a message that indicates you must reload the project. You are prompted to do reload the project in the next step.
8. Click OK to close the message box and then close the 'dotmaster' UserControl TestContainer window.
9. Click Reload.

10. In the Solution Explorer, double-click the new .cs file to display the new custom panel that is generated from the XML file.



Creating a production environment

Upgrade a production client/server environment by installing Datacap software on newly provisioned computers and migrating the application to a new environment.

Procedure

To create a production environment:

1. Install and verify the new version of Datacap software on computers that are not running current Datacap software. For installation details, see [Datacap installation and configuration in a client/server environment](#).
2. Copy the migrated or newly customized applications from your test environment to the newly installed release. For details on migrating database information, see [Copying the application to the Datacap server](#).
3. Test and verify the functions of the new configuration before you deploy it into production.
4. Ensure that users can connect to the newly updated production-ready configuration.
5. Switch to the newly configured system for production use. For example, you might disable user activity permanently on the previous production system and ensure that new server IP addresses are correctly designated.



Uninstalling Datacap

The Datacap installation removal process removes only those files that the previous Datacap installation process created.

Before you begin

Before you begin the uninstall, ensure that all Datacap client and Datacap Web Client users and administrators on all workstations are logged off.

Verify that all developers are logged off and all background client processes are shut down, including Datacap Web Client Upload Service, the Rulerunner Service, Datacap Studio, FastDoc, Maintenance Manager, and Report Viewer.

Finally, ensure that you stop the Datacap Web Client site and any other Datacap sites or web-based services.

After you ensure that no other Datacap software components are running, shut down the Datacap Server Service.

Important: In Windows Explorer, navigate to a network location and create one or more folders for storing the backup copy of your customized application folders and shortcuts.

About this task

Any files, settings, registry keys, administrative users, websites, process model identity changes that you altered or created in Datacap or that were changed by Datacap processing are not removed automatically by

the installation removal process. You must remove these things manually.

If you are installing a newer version of Datacap on the same machine in the same location, customized applications that were untouched by the installation removal process are not changed by the later installation. But you must follow all applicable upgrade instructions, if any, provided with the newer version of Datacap for your custom application.

The complete installation removal process includes several steps.

1. [Backing up custom applications](#)

You must make backups of folders that contain the customization that you want to keep. Make copies of every customized application, including customized Datacap Web Client pages and Report Viewer reports.

2. [Backing up custom application shortcuts](#)

You can make backup copies of the Datacap client shortcuts for custom applications that are found on user and developer workstations or on servers.

3. [Removing Datacap Web Client and Datacap Report Viewer sites](#)

Windows servers on which Datacap Web Client and Report Viewer were installed and configured contain sites that can be removed. The Report Viewer site can be removed manually. The Datacap Web Client Server Configuration tool must be used to unconfigure the Datacap Web Client site before you remove the site.

4. [Removing the Datacap Application Pools](#)

Windows servers on which Datacap Web Client and Report Viewer are configured contain one or more Datacap application pools that you can remove.

5. [Removing IBM Datacap Version 9.0.1](#)

To remove Datacap 9.0.1, use the Microsoft Windows removal process.

6. [Removing accounts from Administrator and Backup Operator groups](#)

A number of Datacap components required that domain/Windows accounts be added to the Administrator or Backup Operator Groups on the server. Repeat this procedure on each server on which Datacap was installed.

7. [Deleting the Datacap registry keys](#)

Every machine on which Datacap components were installed contains Datacap registry keys that you must delete. Repeat this procedure on each machine on which Datacap was installed.

8. [Deleting the program shortcuts](#)

Program shortcuts were created on every computer on which a Datacap component was installed. Repeat this procedure to delete shortcuts from the Windows 7 and Windows 2008 computers on which a Datacap component was installed.

9. [Deleting the remaining folders](#)

Some folders, including the C:\Datacap folder and some of its subdirectories, might not automatically be deleted during the uninstall process. You must manually delete these folders after the uninstall process is completed.

Parent topic: [Installing](#)

Backing up custom applications

You must make backups of folders that contain the customization that you want to keep. Make copies of every customized application, including customized Datacap Web Client pages and Report Viewer reports.

Procedure

To back up your custom applications:

1. Using Windows Explorer on the Datacap server, go to the C:\Datacap folder.

2. Copy the datacap.xml file to your backup folder.
3. Using Windows Explorer, go to C:\Datacap*application name* folder.
4. Copy the *application name* folder to your backup folder.
5. If other files associated with the custom application exist in other locations, copy them to your backup folder as well. For example, your Administrator and Engine databases are on a database server. Your custom Datacap Web Client pages are on the Datacap Web Client Server. Customized Report Viewer reports are on the Report Viewer server.

Backing up custom application shortcuts

You can make backup copies of the Datacap client shortcuts for custom applications that are found on user and developer workstations or on servers.

Procedure

To back up the user shortcuts for custom applications that appear on the Start menu of a Windows workstation or a Windows server.

1. Using Windows Explorer, navigate to the C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Datacap\User*application* folder. Depending on your Windows settings, the C:\ProgramData folder might be hidden. If it is, you can either change the Folder Options to display hidden files, or you can enter the path manually in Windows Explorer.
2. Right-click the *application* folder and select Copy.
3. Using Windows Explorer, navigate to your backup folder.
4. Right-click in the folder and select Paste.

Removing Datacap Web Client and Datacap Report Viewer sites

Windows servers on which Datacap Web Client and Report Viewer were installed and configured contain sites that can be removed. The Report Viewer site can be removed manually. The Datacap Web Client Server Configuration tool must be used to unconfigure the Datacap Web Client site before you remove the site.

Procedure

To unconfigure a Datacap Web Client site.

1. From the Start menu, select IBM Datacap Web > Datacap Web Client Server Configuration Tools. If User Account Control (UAC) is on, the User Account Control window is displayed. Click Yes.
2. Click OK to close the information box.
3. Click Unconfigure.
4. Click OK, then click Exit.

To remove sites and virtual folders:

5. From the Start menu, select Administrative Tools > Internet Information Services (IIS) Manager.
6. Expand the Computer and Sites nodes.
7. Right-click the Datacap Web Client or Report Viewer site and select Remove.
8. Close the Internet Information Services (IIS) Manager window.

Removing the Datacap Application Pools

Windows servers on which Datacap Web Client and Report Viewer are configured contain one or more Datacap application pools that you can remove.

Procedure

To remove Datacap application pools:

1. From the Start menu, select Administrative Tools > Internet Information Services (IIS) Manager.
2. In the Connections pane, select Application Pools. The Application Pools are listed in the center pane.
3. For each Datacap application pool, right-click the application pool and select Remove.

Removing IBM® Datacap Version 9.0.1

To remove Datacap 9.0.1, use the Microsoft Windows removal process.

Procedure

1. In Windows, select Start > Control Panel > Programs > Programs and Features.
2. On the Programs and Features window, scroll down to Datacap 9.0.1 and click Uninstall. A message box prompts you to confirm that you want to remove the software.
3. Click Yes.
4. If the User Account Control (UAC) window is displayed, click Yes.

Removing accounts from Administrator and Backup Operator groups

A number of Datacap components required that domain/Windows accounts be added to the Administrator or Backup Operator Groups on the server. Repeat this procedure on each server on which Datacap was installed.

Procedure

To remove accounts from either or both of the Windows Administrator and Backup Operator groups:

1. From the Start menu, select Administrative Tools > Computer Management > Local Users and Groups > Groups.
2. Double-click the Administrators group to open it and remove all Datacap accounts, then click OK.
3. Double-click the Backup Operators group to open it and remove all Datacap accounts, then click OK.

Deleting the Datacap registry keys

Every machine on which Datacap components were installed contains Datacap registry keys that you must delete. Repeat this procedure on each machine on which Datacap was installed.

Procedure

To delete the Datacap registry keys:

1. From the Start menu, select Run.
2. Enter `regedit`, then click OK. When User Access Control (UAC) is on, click Yes.
3. On both Windows 32 and Windows 64-bit machines, navigate to and delete the following key: `\HKEY_CURRENT_USER\Software\Datacap`.

4. On Windows 32-bit machines, navigate to and delete the following key:\HKEY_LOCAL_MACHINE\SOFTWARE\Datacap.
5. On Windows 64-bit machines, navigate to and delete the following key:\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Datacap.
6. Close the Registry Editor window.

Deleting the program shortcuts

Program shortcuts were created on every computer on which a Datacap component was installed. Repeat this procedure to delete shortcuts from the Windows 7 and Windows 2008 computers on which a Datacap component was installed.

Procedure

1. Using Windows Explorer, navigate to C:\ProgramData\Microsoft\Windows\Start Menu\Programs folder. Depending on your Windows settings, the C:\ProgramData folder might be hidden. If it is, you can either change the Folder Options to display hidden files, or you can enter the path manually in Windows Explorer.
2. Right-click the Datacap folder and select Delete.
3. If there are shortcuts on the desktop, right-click the shortcut, and select Delete.

Deleting the remaining folders

Some folders, including the C:\Datacap folder and some of its subdirectories, might not automatically be deleted during the uninstall process. You must manually delete these folders after the uninstall process is completed.

Procedure

To delete the remaining folders that were not removed by the uninstall process:

1. Using Windows Explorer, navigate to the C:\Datacap folder.
2. Right-click on the Datacap folder and select Delete. If access is denied, delete the individual subfolders before you delete the Datacap folder.
3. If you processed batches using the Datacap Web Client, the Scan Into field identified the folder that was used to temporarily hold images during scanning. The default name and location of this folder is C:\Datacap\scan. If this folder is not in C:\Datacap, navigate to and delete the folder.

Monitorowanie

System można monitorować w trybie online za pomocą Datacap Navigator lub poprzez uruchamianie raportów.

- [Monitorowanie systemu w programie Datacap Navigator](#)
Możesz monitorować system Datacap za pomocą kokpitu w programie Datacap Navigator. Kokpit przedstawia wizualne podsumowanie statystyk systemowych zawierające także alerty dotyczące niektórych metryk systemu, uruchamiane, gdy metryki te nie spełniają konfigurowanych parametrów ustalonego poziomu wydajności.
- [Monitorowanie systemu za pomocą przeglądarki raportów w programie Datacap](#)
Możesz monitorować system Datacap, uruchamiając raporty standardowe lub niestandardowe.

Monitoring your system with Datacap Navigator

You can monitor your Datacap system by using a dashboard in Datacap Navigator. The dashboard shows summary information for the following categories: in-progress tasks, team performance, and OCR accuracy. Included in the system summary are alerts for some system metrics when those metrics do not satisfy configurable service-level-agreement parameters.

The summary information is shown in two main ways:

Donuts	The relative amounts of various items are shown visually in circles that are called <i>donuts</i> . For example, for in-progress tasks, the proportion of batches that are in the Verify task is shown.
Graphs	Item metrics over time are shown.

You can drill down on the summary information to see the details.

- [Accessing the dashboard](#)
You can use the dashboard to monitor your Datacap system.
- [Enabling application statistics collection](#)
To use the dashboard for monitoring accuracy-related statistics, you must enable the collection of statistics for one or more applications.
- [Configuring dashboard features](#)
To use the dashboard, you must add the dashboard as one of your Datacap Navigator desktop features. When you add the desktop as a feature, you enter values for parameters that affect the dashboard display and alerts. Also, you optionally enter an email address for receiving email notifications.

Accessing the dashboard

You can use the dashboard to monitor your Datacap system.

Before you begin

If you did not already do so, perform the following actions:

1. Install Datacap Navigator. For more information, see [Installing and configuring Datacap Navigator](#).
2. Enable statistics collection. For more information, see [Enabling application statistics collection](#).
3. Add the dashboard to Datacap Navigator. For more information, see [Configuring dashboard features](#).

Procedure

To access the dashboard:

1. Open Datacap Navigator. For more information, see [Datacap Navigator access](#).
2. Click the Open Datacap Dashboard icon.

Parent topic: [Monitoring your system with Datacap Navigator](#)

Enabling application statistics collection

To use the dashboard for monitoring accuracy-related statistics, you must enable the collection of statistics for one or more applications.

About this task

Important: Enabling statistics collection for an application makes your application's Export task run slower.

Procedure

To enable statistics collection for an application:

1. In the Start menu, select IBM Datacap Services>Datacap Application Manager. If the User Account Control window opens, click Yes.
2. In the Applications pane, select the application.
3. Click the Main tab.
4. On the Main tab, select Save Statistics.
5. Close the Datacap Application Manager.

Parent topic: [Monitoring your system with Datacap Navigator](#)

Configuring dashboard features

To use the dashboard, you must add the dashboard as one of your Datacap Navigator desktop features. When you add the desktop as a feature, you enter values for parameters that affect the dashboard display and alerts. Also, you optionally enter an email address for receiving email notifications.

The feature name for the dashboard is Datacap Dashboard Page. For the procedure for adding this feature, see [Customizing Datacap Navigator desktops](#). When you select this feature, a Feature configuration pane displays to the right.

- [Enabling email notification](#)
To enable email notification, you must perform some JAR file configuration in addition to specifying an SMTP mail server and email address during dashboard feature configuration.
- [Dashboard parameters](#)
For the SLA JSON string field, you enter parameter values that affect the behavior of the dashboard.

Parent topic: [Monitoring your system with Datacap Navigator](#)

Enabling email notification

To enable email notification, you must perform some JAR file configuration in addition to specifying an SMTP mail server and email address during dashboard feature configuration.

About this task

When you enable email notification, the Datacap system sends emails to alert you of certain conditions based on your configured thresholds. Two types of emails are sent:

Threshold exceeded	<p>An email might be sent when a threshold is reached or exceeded.</p> <p>For example, suppose that you set the pending batch threshold to 60 minutes. If a batch is in pending state for that amount of time or longer, an email might be sent. (An email is sent only if the batch is one of the top 10 pending batches.) The email indicates the exceeded-threshold condition for the batch.</p>
Cleared	<p>Another email is sent when the threshold condition is satisfied.</p> <p>For example, suppose that a threshold-exceeded email was sent for a pending batch. When that batch is no longer pending, an email is sent that indicates the cleared status of the original alert.</p>

Procedure

To configure the JAR files that are needed for email notification:

1. Find the directory that contains the DatacapWebPlugin.jar file on your system. This file is installed as part of the Datacap Navigator installation. For information about this installation, see [Installing and configuring Datacap Navigator](#).
2. Open a DOS command window.
3. In the DOS command window, change directories to the directory that contains the DatacapWebPlugin.jar file. For example, you might change directories to the C:\Datacap\tmlweb.java directory with the following command: `cd C:\Datacap\tmlweb.java`.
4. Run the following commands:

```
jar -xvf DatacapWebPlugin.jar lib\activation.jar
jar -xvf DatacapWebPlugin.jar lib\javax.mail-1.5.4.jar
```

These commands extract the following files:

- o activation.jar
- o javax.mail-1.5.4.jar

5. Copy the extracted files to the following directory: `../navigator.ear/navigator.war/WEB-INF/lib`. This directory is relative to your application server profiles directory for installed applications.

Parent topic: [Configuring dashboard features](#)

Dashboard parameters

For the SLA JSON string field, you enter parameter values that affect the behavior of the dashboard.

For information about these parameters and the specific parameter names to use, see the [Service-level-agreement parameters](#) section in this topic. Use the following format for the field value (but without any carriage returns):

```
{"SLA": {
  "parameter1-name": parameter1-value,
  "parameter2-name": parameter2-value,
  "parameterN-name": parameterN-value
}}
```

Here is an example:

```
{"SLA": {
  "businessName": "IBM Corporation",
  "appName": "TravelDocs",
  "batchesAbortedInPresetTime": 20,
  "batchesPendingInPresetTime": 100,
  "pageAccuracy": 97.9,
  "fieldAccuracy": 96.5
}}
```

Service-level-agreement parameters

Parameter subject matter	Parameter name	Statistics type	Comments
--------------------------	----------------	-----------------	----------

Parameter subject matter	Parameter name	Statistics type	Comments
Accuracy rate - field - minimum	fieldAccuracy	OCR accuracy	A percentage such as 96.5 that specifies the minimum OCR accuracy rate for fields. A warning is shown if the recent OCR accuracy rate is below the specified minimum but not more than 10% below the minimum. An alert is shown if the recent OCR accuracy rate is more than 10% below the minimum.
Accuracy rate - page - minimum	pageAccuracy	OCR accuracy	A percentage such as 97.9 that specifies the minimum accuracy rate for page type identification. A warning is shown if the recent page identification accuracy rate is below the specified minimum but not more than 10% below the minimum. An alert is shown if the recent page identification accuracy rate is more than 10% below the minimum.
Batches - aborted	batchesAbortedInPresetTime	In-progress tasks	<p>The number of minutes that a batch must have an aborted status to be reported by the dashboard as aborted. For example, a parameter value of 60 means that the dashboard shows all aborted batches that were aborted 60 minutes ago or longer.</p> <p>If you configure email notification, an email is sent for a batch in the following circumstances: the batch has an aborted status beyond the specified period, and the batch is one of the top 10 longest aborted batches. For details about the emails that are sent, see Enabling email notification.</p>
Batches - pending	batchesPendingInPresetTime	In-progress tasks	<p>The number of minutes that a batch must have a pending status to be reported by the dashboard as pending. For example, a parameter value of 120 means that the dashboard shows all pending batches that were made pending 120 minutes ago or longer.</p> <p>If you configure email notification, an email is sent for a batch in the following circumstances: the batch has a pending status beyond the specified period, and the batch is one of the top 10 longest pending batches. For details about the emails that are sent, see Enabling email notification.</p>

Parameter subject matter	Parameter name	Statistics type	Comments
Name - application	appName	N/A	The names of the applications for which you enabled statistics collection. For information about enabling statistics collection, see Enabling application statistics collection .
Name - business	businessName	N/A	The name of your business

Parent topic: [Configuring dashboard features](#)

Monitoring your system with Datacap Report Viewer

You can monitor your Datacap system by running standard or custom reports.

- [Logging in to Datacap Report Viewer](#)
When you log in to Report Viewer, the user ID that you use on the login page is automatically logged in to all Datacap applications for which that user ID is a valid Datacap user.
- [Viewing a standard Datacap Report Viewer report](#)
You can run and view standard Report Viewer reports to get batch productivity results.
- [Viewing a custom report in the Datacap Report Viewer web interface](#)
You can review a custom report in the Report Viewer web interface. If you are logged on to the Report Viewer home page, you must log off and log back in to see the new report.
- [Creating a Datacap Report Viewer report filter](#)
You can filter the data that is displayed in a Report Viewer report, based on the values in one or more fields. For example, if you want to view activity that is related to a specific operator, you filter the data by operator name.
- [Adding reports to the Datacap Report Viewer dashboard](#)
You can view up to six reports simultaneously on the Report Viewer dashboard. The dashboard is typically used for viewing chart-based reports.
- [Modifying the Datacap Report Viewer dashboard](#)
You can re-size, move, or remove any of the reports that are displayed on the dashboard.
- [Standard reports](#)
Report Viewer provides a set of standard reports and the ability to customize existing reports and create new reports.

Logging in to Datacap Report Viewer

When you log in to Report Viewer, the user ID that you use on the login page is automatically logged in to all Datacap applications for which that user ID is a valid Datacap user.

Before you begin

If you did not already do so, install Datacap Report Viewer. For information about a single-machine installation, see [Datacap Report Viewer](#). For information about a client/server installation, see [Installing and configuring Datacap Report Viewer](#).

About this task

The following procedure describes how to access and log in to Report Viewer in either a single computer environment or when you are running Internet Explorer on the Report Viewer Web Server.

Procedure

To log in to Report Viewer:

1. Verify that the Datacap Server service is started. For more information, see [Ensuring that the Datacap Server Service is started](#).

2. Open Internet Explorer, and connect to:
Single machine environment (local host):

`http://127.0.0.1/RV2/Login.aspx`

Remote web server:

`http://WebServer/RV2/Login.aspx`

3. Enter the Datacap Admin User, Password, and Station (admin, admin, and 1), then click Login The Report Viewer home page is displayed with the Datacap applications to which the Datacap login user ID has access.

Parent topic: [Monitoring your system with Datacap Report Viewer](#)

Viewing a standard Datacap Report Viewer report

You can run and view standard Report Viewer reports to get batch productivity results.

Procedure

To view a standard Report Viewer report:

1. On the Report Viewer home page, select the Batch Productivity report.
2. Select the application and click Run Report.
3. Click Reports to return to the Report Viewer home page.

Parent topic: [Monitoring your system with Datacap Report Viewer](#)

Viewing a custom report in the Datacap Report Viewer web interface

You can review a custom report in the Report Viewer web interface. If you are logged on to the Report Viewer home page, you must log off and log back in to see the new report.

Before you begin

Custom reports must be created.

Procedure

To view a custom report in the Report Viewer web interface:

1. In your web browser, go to the Report Viewer home page, `http://<machine_name>/Login.aspx`.
2. Enter your user name, password, and station ID. Click Login.
3. Select your application from the application list. You can select multiple applications by holding down the CTRL key.

Important: The application list comes from the file C:\Datacap\datacap.xml. If an application that you are looking for is not listed, verify that the application is included in the datacap.xml file.

4. Select your custom report.

Important: If your report is not listed, log out of Report Viewer and log back in again. You must log out and log back in any time that you add a report or modify an existing report.

5. Click Run Report. Report Viewer displays the default custom report.

Parent topic: [Monitoring your system with Datacap Report Viewer](#)

Creating a Datacap Report Viewer report filter

You can filter the data that is displayed in a Report Viewer report, based on the values in one or more fields. For example, if you want to view activity that is related to a specific operator, you filter the data by operator name.

About this task

Each filter is associated with a single report. The Select Filter list on the Report Viewer home page displays only those filters that are available for the report that you selected.

Procedure

To create a Report Viewer report filter:

1. On the Report Viewer home page, select the report from the list.
2. Click Manage Filters to display the Filter-related fields.
3. In the Add Filter field, enter a name for the new filter.
4. Click Add to display more fields.
5. Set up the filter as follows:
 - a. In the first field, select the column name such as Operator.
 - b. In the second field, select the action such as equal to.
 - c. In the third field, enter the value such as admin.
6. Click Add Filter and repeat the previous step to add another field.
7. Select Public to make the filters available to all users or Private to make the filters available only to you.
8. Click Save.
9. Click Run Report to run the report by using the new filter.
10. Click Reports to return to the Report Viewer home page.

Parent topic: [Monitoring your system with Datacap Report Viewer](#)

Adding reports to the Datacap Report Viewer dashboard

You can view up to six reports simultaneously on the Report Viewer dashboard. The dashboard is typically used for viewing chart-based reports.

Procedure

To add reports to the Report Viewer dashboard:

1. On the Report Viewer home page, click Dashboard.
2. Select the report and the application name to display the report. If applicable, select a filter.
3. To add another report, click Add and repeat the previous step.
4. If you want the data on the reports to refresh automatically, click Refresh and select a refresh interval.
5. Click Reports to return to the Report Viewer home page.

Modifying the Datacap Report Viewer dashboard

You can re-size, move, or remove any of the reports that are displayed on the dashboard.

Procedure

To modify the Report Viewer dashboard:

1. On the Report Viewer home page, click Dashboard.
2. To re-size a report, click the cross-hatched area on the window and drag.
3. To move a report, click the dotted area of the report and drag the report to a different location.
4. To remove a report, click the report that you want to remove, then click Remove.
5. Click Reports to return to the Report Viewer home page.

Parent topic: [Monitoring your system with Datacap Report Viewer](#)

Standard reports

Report Viewer provides a set of standard reports and the ability to customize existing reports and create new reports.

The following standard reports for monitoring batch status, station activity, and problem batches are included with the Report Viewer software component:

Table 1. Standard reports for monitoring batch status, station activity, and problem batches

Report Name	Description	Data source
Problem Batches	Displays all queued batches that aborted or are offline, or ran for longer than 2 hours.	queue table
Current Batches	Displays all queued batches that are not complete or were not canceled.	queue table
Batch Aging	Same as Current Batches except that it includes elapsed time since batch was started.	queue table
Current Stations	Displays the station IDs and IP addresses of all stations that are currently logged, plus the number of batches that are processed by each station.	qstats table

Report Name	Description	Data source
Station Activity	Displays a bar chart that shows the number of batches that are completed by each station within the last 4 hours.	qstats table
Pending Tasks	Displays a bar chart that shows the number of queued batches with a status of Pending.	queue table
Tasks Use Hourly	Displays a bar chart for the selected date range that shows the percentage of each hour spent processing by each station.	taskstats table
Tasks Use Daily	Displays a bar chart for the selected date range that shows the percentage of each day spent processing by each station.	taskstats table
Tasks	<p>Background Displays summary of all batches that are processed, grouped by station ID. Includes aggregates for processing time, number of pages that are scanned, and so on, for each station.</p> <p>Scan Displays a list of the Datacap Desktop scanned batches, which are grouped by operator. Includes aggregates for processing time, number of pages that are scanned, and so on, for each operator.</p> <p>Fixup Displays a list of all batches that were through Fixup, which is grouped by operator. Includes aggregates for processing time, number of pages that are rescanned, inserted, and so on, for each operator.</p> <p>Verify Displays all batches that are in or ready for verification, which is grouped by operator. Includes aggregates for processing time, pages processed, average documents per hour, and so on, for each operator.</p>	taskstats table
Tasks Summary	<ul style="list-style-type: none"> • Background Summary • Scan Summary • Fixup Summary • Verify Summary 	taskstats table
Batch Productivity	Displays a list of all batches with status of "Job done." Includes number of pages that are scanned and total elapsed time for each batch from when batch was started until completed. Also includes average number of pages and average elapsed time per batch.	queue and tmbatch tables
Debug	Displays a list of all batches in the debug table (for example, batches that were deleted manually). Includes the batch status, status before the batch is moved to the debug table, and so on.	debug table

Dostosowywanie i uruchamianie aplikacji

Razem z produktem IBM Datacap dostarczane są przykładowe aplikacje branżowe.

- [Używanie pulpitu aplikacji Datacap z aplikacjami Datacap](#)
Po pomyślnej instalacji programu Datacap w wersji 9.0.1 możesz wykorzystać program Datacap Desktop do tworzenia partii, rozpoznawania stron, tworzenia dokumentów i eksportowania partii.
- [Przygotowywanie i uruchamianie aplikacji Fastdoc](#)
W programie FastDoc możesz tworzyć aplikacje do skanowania, indeksowania i ręcznego uruchamiania czynności dotyczących dokumentów w tle zarówno w autonomicznym środowisku, jak i jako klient programu Datacap.
- [Uruchamianie aplikacji Datacap Accounts Payable](#)
Użyj aplikacji klienckiej programu Datacap Accounts Payable do przetwarzania przykładowych, wstępnie zeskanowanych obrazów faktur zawartych w opcji produktu Datacap Accounts Payable.
- [Zastosowania dla wniosków medycznych](#)
Użyj aplikacji klienckiej programu Datacap Medical Claims do automatyzacji wprowadzania danych z profesjonalnych formularzy wniosków (CMS 1500) używanych przez indywidualnych świadczeniodawców lub dostawców usług medycznych oraz formularzy wniosków (UB04) używanych przez instytucje, na przykład szpitale.
- [Aplikacja Traveldocs](#)
Przykładowa aplikacja TravelDocs wchodząca w skład instalacji produktu Datacap pokazuje, w jaki sposób aplikacja Datacap przetwarza różne dokumenty związane z podróżą.
- [Konfigurowanie składnika Rulerunner do wykonywania czynności w aplikacji](#)
Po pomyślnym skonfigurowaniu aplikacji do ręcznego przetwarzania czynności możesz skonfigurować składnik Rulerunner w celu automatycznego uruchamiania w czynności aplikacji w tle. Niektóre z czynności, dla których możesz zechcieć skonfigurować uruchamianie składnika Rulerunner, obejmują rozpoznawanie, wstępne przetwarzanie obrazu, sprawdzanie poprawności i eksportowanie. Czynności skanowania wirtualnego (VScan) można także uruchomić, jeśli są przygotowane do automatycznego gromadzenia obrazów bez konieczności ręcznego wyboru obrazów przez użytkownika.

Running tasks with Datacap Desktop

You can use Datacap Desktop to run application tasks.

You can confirm that a task is complete and that the next task is pending by checking the Job Monitor in Datacap Desktop, FastDoc, or your web client.

- [Task profiles for Datacap Desktop](#)
You can use Datacap Desktop to run various tasks. Some of these tasks are automated, which means that they do not require human intervention.
- [Starting a task with Datacap Desktop](#)
You can use Datacap Desktop to run all of the default tasks that the Datacap Studio application wizard generates.
- [Running the VScan task with Datacap Desktop](#)
The VScan task creates a batch by importing electronic files from a specified location. It is mostly used for demonstration purposes.
- [Running the Scan task with Datacap Desktop](#)
You can use Datacap Desktop to scan hardcopy pages.
- [Running the Fixup task with Datacap Desktop](#)
The Fixup task requires user input to correct any errors or integrity issues that were encountered in a preceding task. The batch does not continue to the next task until the Fixup task is completed. Datacap

Desktop identifies the problem document, page, field, or subfield in the batch so that you can make any corrections.

Related information:

[Job monitoring](#)

Task profiles for Datacap Desktop

You can use Datacap Desktop to run various tasks. Some of these tasks are automated, which means that they do not require human intervention.

Tasks are instances of task profiles. The following table shows the relevant task profiles for Datacap Desktop.

Task profile	Automated task?	Description
VScan	No	Creates a batch by importing electronic files from a specified location. For more information, see Running the VScan task with Datacap Desktop .
Scan	No	Scans hardcopy pages. For more information, see Running the Scan task with Datacap Desktop .
PageID	Yes	Identifies the page type of a scanned image. Specifically, Datacap Desktop applies the page type identification rules on a batch and displays a completion message when the task is finished.
Profiler	Yes	Creates documents out of the pages. To do so, Datacap Desktop applies rules that are based on predefined criteria. Datacap Desktop displays a completion message when the task is finished.
Fixup	No	Fixes errors or problems. For more information, see Running the Fixup task with Datacap Desktop .
Verify	No	Verifies the accuracy of the captured data.
Export	Yes	Exports a pending batch to a specified location. Datacap Desktop displays a completion message when the task is finished. Important: You must complete the Verify task, if it is included in the job, before you can complete the Export task. The Verify task can be completed only in the web client or Datacap Desktop, depending on how you configured the Verify task.
Background	Yes	Automates the completion of pending batches and completes all automated tasks. The automated tasks are run consecutively, except for the VScan, Fixup, and Verify tasks.

Parent topic: [Running tasks with Datacap Desktop](#)

Starting a task with Datacap Desktop

You can use Datacap Desktop to run all of the default tasks that the Datacap Studio application wizard generates.

About this task

Datacap applications include tasks that you can run from the Datacap Web Client or Datacap Desktop. The client that you use to run a task depends on your business requirements and environment.

Procedure

To start a task:

1. Start Datacap Desktop. In the Start menu, select IBM Datacap Clients > Datacap Desktop. Select the application for the task that you want to run, and log on with an account that has sufficient rights to complete the task.
2. Select the task from the Shortcut menu and click Start.
Tip: If you do not see the task in the Shortcut menu, confirm that you specified Datacap Desktop as the program to use in the task's Setup dialog window in the Datacap Web Client. For more information, see [Creating and configuring a task to use with Datacap Desktop](#).

Parent topic: [Running tasks with Datacap Desktop](#)

Running the VScan task with Datacap Desktop

The VScan task creates a batch by importing electronic files from a specified location. It is mostly used for demonstration purposes.

Before you begin

Optionally configure barcode recognition. For more information, see [Setting up a barcode type as a document separator](#).

Procedure

To run the VScan task:

1. In Datacap Desktop, start the VScan task. In the Scan from field, browse to the location from which to import the files and then click Scan. For more information, see [Starting a task with Datacap Desktop](#).
2. If necessary, arrange the batch by removing images or moving the images up or down to change their order.
3. Click Done.

Parent topic: [Running tasks with Datacap Desktop](#)

Running the Scan task with Datacap Desktop

You can use Datacap Desktop to scan hardcopy pages.

Before you begin

- Set up your scanner. For more information, see [Setting up your scanner](#).
- Optionally configure barcode recognition. For more information, see [Setting up a barcode type as a document separator](#).

Procedure

To scan hardcopy pages:

1. Load the pages that you want to scan into your scanner's feeder.
2. In Datacap Desktop, start the Scan task. Log on with an account that has sufficient rights to complete a scanning task. For more information, see [Starting a task with Datacap Desktop](#).

3. Click Select, if you are scanning documents from Datacap Desktop for the first time or if you are changing your scanner.
4. After you select your scanner, click Configure, if you want to change your scanner's settings. The settings that you can modify are determined by the scanner's options and function.
5. After you modify the scanner settings, click OK.
Tip: Datacap Desktop provides the option for specifying the paper source and the scanned image's color mode, DPI resolution, brightness, and contrast.
6. Click Scan.
 - Datacap Desktop arranges the thumbnails of the scanned images in the Image View pane, which you can arrange by clicking Off, Vert, or Horz.
Tip: You can rotate an image by selecting the image in the Batch View pane and clicking one of the rotation buttons in the center pane.
 - The Batch View pane displays a list of scanned images in the current batch. You can organize the scanned images by selecting an image and then clicking the corresponding button to insert, replace, remove, move up or down, join, or split the image or document.
Tip: The buttons in Datacap Desktop are enabled with tooltips that provide information about the function of the button.
7. If necessary, you can place more pages into your scanner's feeder and click Scan to add images to the batch.
8. After you finish scanning, select an image in the Batch View pane. Datacap Desktop displays the selected image in the Image View pane and assigns the page type by selecting an option from the Type menu.
9. In the Status field, select either OK or Problem from the menu.
Important: Selecting the Problem value places the batch in a Pending status and the batch does not proceed to the next task in the workflow until a Fixup task is successfully completed.
10. After you finish organizing the batch and assigning page types and statuses, click OK. Datacap Desktop displays a message that the batch is finished and that you can proceed to the next batch.
11. You can confirm that the scan task is complete and that the next task is pending by checking the Job Monitor in Datacap Desktop, FastDoc, or your web client.

Parent topic: [Running tasks with Datacap Desktop](#)

Running the Fixup task with Datacap Desktop

The Fixup task requires user input to correct any errors or integrity issues that were encountered in a preceding task. The batch does not continue to the next task until the Fixup task is completed. Datacap Desktop identifies the problem document, page, field, or subfield in the batch so that you can make any corrections.

About this task

In Fixup task, you can manipulate, rearrange or delete pages. The pages are marked in the Verify task. However, the actual operation is completed in the Fixup task. You might need to rescan an image as part of the Fixup task. There are two possible selection modes for rescanned images: automatic and manual. For more information, see [Configuring the image selection mode for the Datacap Desktop Fixup task](#).

Procedure

To rescan images in manual selection mode:

1. Scan the page one or more times. For example, you might rescan a page three times so that you have a total of four images for that page. All of the rescanned images are stored in the batch folder until you submit the batch.

2. Navigate to the image that you want to retain. To do so, use Ctrl+U to move backwards through the images and Ctrl+Y to move forwards.
3. Submit the batch. The current image is retained and all other images are deleted.
Important: If you put a batch on hold and later return to that batch, the current image is initially the last scanned page image. That is, the system does not save the current page image when you put the batch on hold. Therefore, you might need to navigate back to the image that you want to retain.

Parent topic: [Running tasks with Datacap Desktop](#)

Preparing and running FastDoc applications

You can create applications on FastDoc to scan, index, and manually run background tasks on documents in a stand-alone environment or as a client to Datacap.

In Local mode, you can use FastDoc for rapid application development with the Datacap application as a starting point to quickly create new Datacap applications without using Datacap Studio.

In Datacap Server mode, FastDoc runs as a client that scans, auto indexes, and uploads batches of documents to Datacap Server. FastDoc is extended to take advantage of Datacap Studio, Application Manager, and other Datacap features.

In Datacap Server mode, you can create an application and setup document hierarchy items such as batches, documents, pages, and fields for that application. You can configure compiled rulesets for the document hierarchy items to run functions like import, page identification, image enhancement, fingerprint matching, and export on the application. The compiled rulesets support inheritance and automatic binding to the document hierarchy items.

Compiled rulesets are self contained application building blocks that you can assemble into an application and configure by using FastDoc or Datacap Studio. These rulesets reduce the expertise and complexity that is needed to create applications by standardizing how core functions are implemented. They also make applications more consistent and easier to understand and support.

Some of the common Datacap features that are not implemented in FastDoc include.

- Line items
- Raising task conditions
- Alternative task profiles
- SELECT and Lookup DCO properties

In Datacap Server mode, FastDoc can use the Datacap Server to further develop the application by using Datacap Studio and Application Manager.

- [Getting started on FastDoc](#)
The first time that you run FastDoc, you must create a Datacap application and add documents, pages, and fields to it.
- [Application configuration on FastDoc](#)
You can configure your application for FastDoc by setting up batch profiles and adding rulesets to the FastDoc workflow.
- [Document processing on FastDoc](#)
FastDoc automates the capture of index data from computer or manually printed documents, which eliminates manual data entry. You can create batches of documents, run indexing and verification tasks, and export the index data and document content to a storage location.

Getting started on FastDoc

The first time that you run FastDoc, you must create a Datacap application and add documents, pages, and fields to it.

After you create the application, you configure settings and rulesets for the documents, pages, and fields that you added. You can then configure import and export rules at the batch level of the application.

- [Starting FastDoc](#)
When FastDoc is installed, program shortcuts for both Administrators and Operators are created on your Windows Start menu. Select the shortcut that is appropriate for you.
- [Creating an application in Application Wizard](#)
You can use the Application Wizard in FastDoc to create a Datacap application. This method is an alternative to using an application that was created by the Forms or Learning template.
- [Forms template application configuration](#)
Use the Forms template to pre-configure the application to process forms on which you know the types of data that you want to capture and where that data is on each image.
- [Learning template application configuration](#)
You can configure learning template applications to process forms with data in both predefined and unknown locations.
- [Configuring export options](#)
The Export task in the workflow for your job contains an Export ruleset that exports documents to a file system. You can add rulesets to export documents to the IBM® Content Manager, IBM FileNet® Content Manager, and Box repositories for storage.

Parent topic: [Preparing and running FastDoc applications](#)

Starting FastDoc

When FastDoc is installed, program shortcuts for both Administrators and Operators are created on your Windows Start menu. Select the shortcut that is appropriate for you.

About this task

If you are using a scanner, start your scanner before you start FastDoc.

You can run FastDoc in the following modes:

- Local mode as a stand-alone client that scans, manually indexes, and uploads batches to repositories and files without using Datacap Server or other Datacap components.
- Datacap Server mode as a client for Datacap applications that scans, auto indexes, and uploads batches to the Datacap Server.
- Local and Datacap Server mode to run batches locally and upload them to Datacap server in the background.

Procedure

To start the FastDoc client:

1. From the Windows Start menu, select one of the following options.
 - a. Operators, select All Programs > IBM Datacap Clients > Datacap FastDoc.
 - b. Administrators, select All Programs > IBM Datacap Developer Tools > Datacap FastDoc (Admin).
2. For Local mode, select Local and click Login.
3. For Datacap Server mode, log in to Datacap Server.
 - a. Select the Datacap application that you want to run in FastDoc.
 - b. Enter a valid user ID and Password for the workstation.

- c. Enter the ID of the workstation.
 - d. Click Login.
4. If you try to exit after you start FastDoc, your batch is put on hold before exiting. If FastDoc is processing a batch when you try to exit, FastDoc completes this processing before you can exit.

Parent topic: [Getting started on FastDoc](#)

Creating an application in Application Wizard

You can use the Application Wizard in FastDoc to create a Datacap application. This method is an alternative to using an application that was created by the Forms or Learning template.

About this task

You can configure the Datacap application that you create to scan, index, and export batches of documents to repositories, file systems, and Datacap Server.

Procedure

To create an application in Application Wizard:

1. Start FastDoc in Local mode.
2. At the FastDoc main window, click Application Wizard.
3. On the Overview window, click Next.
4. Select Create a new RRS application or Create a new CMIS based application and click Next.
5. Enter a name for the application and select an application template.
 - a. Select FormTemplate to use the application for structured images. When you know the types of data that you want to capture and where that data is on each image, select the Form application. For example, a 1040EZ tax form and the types of data on the form, such as name and address, are in the same location on every 1040EZ form. The Form application sets up a workflow that you can match against your fingerprints.
 - b. Select LearningTemplate to use the application for unstructured images. When you know the types of data that you want to capture but you do not know where that data is contained in the image because the location of the data is different on each image, select the Learning application. For example, if you want to capture the date, amount, and tax for expenses from different hotels, the receipt images from each hotel are unique. The location of the data that you want to capture differs for each hotel receipt image so the data cannot be identified with Datacap fingerprints. The Learning application template sets up a workflow where you can add rules, such as Locate rules, for Datacap to learn the different hotel receipt formats as they are encountered.
6. Click Next.
7. Enter the CMIS connection information if you selected Create a new CMIS based application.
8. At the Document hierarchy window, click Next.
9. At the Add sample images window, click + and add the images that you want to process to the C:\Datacap\appname\Images\Input directory.
10. Click Finish to begin the creation process.
11. Check Connect to application when wizard exits to create the application.
12. Click Close to close the Application Wizard.
13. Repeat these steps for both templates if they are needed.
14. Click Local and select Logout to exit Local mode.

Parent topic: [Getting started on FastDoc](#)

Forms template application configuration

Use the Forms template to pre-configure the application to process forms on which you know the types of data that you want to capture and where that data is on each image.

Complete the following tasks to configure Forms template applications.

1. Add the document types, page types, and field types that are needed to replicate the form that you want to capture.
 2. Set up your fingerprints and zone the fields on the pages.
 3. Set up data recognition on each field by using the Recognize Pages and Fields rule set.
 4. Set up validation characteristics on each field by using the Validate Fields ruleset.
 5. Configure how and to which repository you want to export your documents by using one of the Export rulesets.
- [Jobs available in Forms template applications](#)
The following jobs are available in Forms template applications.
 - [Setting up documents on Forms template applications](#)
When you use applications built from the Forms template, you must set up the document types, page types, and field types that are needed to replicate the form that you want to capture. The next time that the document is scanned, FastDoc recognizes the document, and automatically populates data in the fields that you set up.
 - [Setting up image enhancement on Forms template applications](#)
You configure image enhancement at the page level on Forms Template applications to remove lines, shading, misalignment, and other imperfections that can interfere with the recognition process.
 - [Setting up fingerprints on Forms template applications](#)
You set up fingerprints on applications created from the Forms template that correspond to the field layout of the forms that you want to process.
 - [Setting up field recognition on Forms template applications](#)
You configure recognition on the field level of Forms template applications to recognize the fields that you want to capture and convert those fields into data.
 - [Setting up field validation on Forms template applications](#)
You configure validation on the field level of Forms template applications to determine whether the captured data complies with the data integrity rules that are defined in your business requirements.

Parent topic: [Getting started on FastDoc](#)

Jobs available in Forms template applications

The following jobs are available in Forms template applications.

About this task

DemoSingleTIFFS

Imports single-page TIFF images from the file system using rules.

Web Job

Use this job if you are scanning using Datacap Web Client and manually selecting files from the file system.

Navigator Job

Use this job if you are scanning using Datacap Navigator and manually selecting files from the file system.

Manual Select

Use this job if you are scanning using Datacap Desktop or FastDoc and manually selecting files from the file system.

DemoMultiFormat

Imports any supported file type from the file system using rules. This job is similar to DemoSingleTIFFS, but allows for all supported file types in addition to TIFF images. All files are converted to single-page TIFF files as part of processing. Supported file types include:

- Multipage TIFF images
- Bitmap images
- ZIP archives
- PDF
- Microsoft Word, Outlook, and Excel documents
- HTML
- Text files (RTF or TXT)

Fixup

If a batch is found to have document integrity problems, the batch is routed to the Fixup job, which allows you to classify the page in the document hierarchy or remove it from the batch.

VerifyExport

If documents pass validation in your workflow, they are sent directly to the Export task. However, if a document does not pass validation (for example: due to low confidence in the recognition of text in a field), then the document is routed instead to the VerifyExport job, where you can verify the recognition and correct it if needed. Once you have verified the data, it can then be exported to your repository.

Parent topic: [Forms template application configuration](#)

Related tasks:

[Configuring export options](#)

Setting up documents on Forms template applications

When you use applications built from the Forms template, you must set up the document types, page types, and field types that are needed to replicate the form that you want to capture. The next time that the document is scanned, FastDoc recognizes the document, and automatically populates data in the fields that you set up.

Procedure

To set up documents on Forms template applications:

1. Start FastDoc in Datacap Server mode.
2. Log in to your application created from the Forms template.
3. Click the Configure documents, pages, and fields icon.
4. In the Batch Structure pane, select the Batch name, such as Forms, and click Add Document. The Batch name is the same as the application name.
 - a. Enter a unique name for the Document Type. For example: `Tax Form`.
 - b. Check Enable for Use rulesets from and select Document. The document rulesets will be run whenever this new document type is encountered.
 - c. Click Add.
 - d. Add the remaining documents that you want to include in this batch.
 - e. Click Save.
5. Select a new document and click Add Page.
 - a. Select Create New to create a new page type and enter a page name. For example: `Tax Form Page 1`.
 - b. Check Enable for Use rulesets from and select Page. The page rulesets will be run whenever this new document type is encountered.

- c. Select Use Existing if you want to use a page type that is already defined.
 - d. Click Add.
 - e. Add the remaining pages that you want to include in the document. For example: `Tax Form`
`Signature Page`.
 - f. Click the Settings tab for each page and configure how many of the page type can be in the document and how to position the page type in the document. These settings are used to split pages into separate documents. If you set Maximum to 0, there is no limit on the number of page types in the document. Setting both Maximum and Minimum to 0 allows the page to be either present or missing with no restrictions. Setting both Maximum and Minimum to 1 will cause a new document to be created every time the page is encountered.
 - g. Click Save.
6. Select a new page and click Add Field.
 - a. Select Create New to create a new field type and enter a field name. For example: `Last Name`.
 - b. Select Use Existing if you want to use a field type that is already used in another document.
 - c. Click Add.
 - d. Add the remaining fields that you want to include on the page of the document.
 - e. Click the Settings tab for each field and configure the field settings by using the hover help text for each field as a guide.
 - f. Click Save.

Parent topic: [Forms template application configuration](#)

Related tasks:

[Configuring rulesets for the application](#)

[Testing rulesets](#)

Setting up image enhancement on Forms template applications

You configure image enhancement at the page level on Forms Template applications to remove lines, shading, misalignment, and other imperfections that can interfere with the recognition process.

Procedure

To setup image enhancement on Forms template applications:

1. Start FastDoc in Datacap Server mode.
2. Log in to the application you created using the Forms template.
3. Click the Configure documents, pages, and fields icon.
4. Select the page named Other in the Batch Structure pane and click the Ruleset tab.
5. Select the Image Enhancement ruleset.
6. Click Open an image file and select the image.
7. Select the enhancement options that you want to run to clean up the image. These options are run when you process the image at run time.

Parent topic: [Forms template application configuration](#)

Setting up fingerprints on Forms template applications

You set up fingerprints on applications created from the Forms template that correspond to the field layout of the forms that you want to process.

About this task

When a zone for a field is created in the wrong place or is the wrong size, the index data for that field is not picked up in the Profiler task.

Procedure

To create fingerprints on Forms template applications:

1. Start FastDoc in Datacap Server mode.
2. Log in to your application created from the Forms template.
3. Click the Configure documents, pages, and fields icon.
4. In the Batch Structure pane, select the page for which you want to create fingerprints and click the Fingerprints tab.
5. In the Fingerprint Class field, select the default <New> fingerprint class. A list of the fingerprints that are assigned to the fingerprint class is displayed in the Fingerprint field.
 - a. Click Add to create a new fingerprint class. Enter the name of the fingerprint class and click Add. You can name the class to correspond with the name of the client who sent the form to you.
 - b. Select a fingerprint class and click Delete to delete a fingerprint class.
 - c. Select a fingerprint and click Rename to change the name of an existing fingerprint class.
 - d. Click Reload to refresh the list of fingerprint classes.
6. In the Fingerprints section:
 - a. Click Add and go to the folder that contains the image to be used to create the fingerprint. For example, C:\Datacap\APT\Images\Input.
 - b. Select the image and click Open. The fingerprint is added to the list of fingerprints.
 - c. Click a fingerprint to display the fingerprint image on the Image tab.
 - d. Double-click a fingerprint and use drag and drop to set the zones on the image.
 - e. Click Zoom In or Zoom Out to enlarge or reduce the size of the fingerprint image.
 - f. Select a fingerprint and click Delete to delete a fingerprint.
 - g. Select a fingerprint and click Change class to to associate the fingerprint with another fingerprint class.
 - h. Select a fingerprint and click Change page type to to associate the fingerprint with another page type.
7. Create fingerprint zones for the fields on the fingerprint file. If you want to draw zones on fields that contain zones, you must draw the smaller zones first. Then, you draw the zones in the parent fields that contain the smaller zones.
 - a. Select the field in the Batch Structure pane.
 - b. Click Zones on the Image tab.
 - c. Click the field on the image and drag the cursor to draw the zone. The Position field displays the coordinates of the fingerprint zone.
 - d. Click the edge of the zone and drag the straight arrow icon to resize the zone.
 - e. Click the zone and drag the crossed arrows icon to move the zone.
 - f. Highlight a fingerprint zone and click Clear Zone to delete fingerprint zone positions.

Attention: You can update field zones from the Test tab while you are testing the rulesets and the images.

Tip: If the wrong fingerprint was matched and you want to keep the existing fingerprint, select Add to create a new fingerprint and retain the existing fingerprint. If the format of the form was changed since the fingerprint was created and the correct fingerprint was matched, select Delete to remove the existing fingerprint. Then, click Add to create a fingerprint with the new format to replace the one that you deleted.

Parent topic: [Forms template application configuration](#)

Setting up field recognition on Forms template applications

You configure recognition on the field level of Forms template applications to recognize the fields that you want to capture and convert those fields into data.

Procedure

To set up field recognition on Forms template applications:

1. Start FastDoc in Datacap Server mode.
2. Log in to the application created using the Forms template.
3. Click the Configure documents, pages, and fields icon.
4. Select the field in the Batch Structure pane and click the Ruleset tab.
5. Select the Recognize Pages and Fields ruleset.
6. Check the Read Zones field.
7. Select the recognition options that you want to run by using the hover text help as a guide. These options are run when you process the image at run time.
8. Run these steps on every field for which you want to run recognition.

Parent topic: [Forms template application configuration](#)

Setting up field validation on Forms template applications

You configure validation on the field level of Forms template applications to determine whether the captured data complies with the data integrity rules that are defined in your business requirements.

Procedure

To set up field validation on Forms template applications:

1. Start FastDoc in Datacap Server mode.
2. Log in to the application created using the Forms template.
3. On the Datacap window, click the Configure workflow icon and select the job that you want to run.
4. Click the Configure documents, pages, and fields icon.
5. Select the field in the Batch Structure pane and click the Ruleset tab.
6. Select the Validate Fields ruleset.
7. Check Validate this field.
8. Configure the validation options that you want to run by using the hover text help as a guide. These options are run when you process the image at run time.
9. Run these steps on every field for which you want to run validation.

What to do next

Important: Make sure your validations match your export. For example, if you are exporting to a database column of type integer, set the validation to make sure the value being exported is an integer, and ensure that the validation cannot be overridden.

Parent topic: [Forms template application configuration](#)

Learning template application configuration

You can configure learning template applications to process forms with data in both predefined and unknown locations.

The Learning template uses generic document and page types that are provided with the template. You do not have to set up fingerprints for applications that are built on this template.

Complete the following tasks to configure Learning template applications.

1. Add the field types, in addition to the ones that are provided in the template, that you want to capture.
 2. Configure the Recognize Pages and Fields ruleset to find words that identify the position of data on a page, such as Date, Order Number, and Total.
 3. Set up validation characteristics on each field by using the Validate Fields ruleset.
 4. Configure how and to which repository you want to export your documents by using one of the Export rulesets.
- [Jobs available in Learning template applications](#)
The following jobs are available in Learning template applications.
 - [Setting up fields on Learning template applications](#)
The Learning template uses generic document types, page types, and field types that are provided with the template. You can add other files types that you want to capture with your application.
 - [Setting up image enhancement on Learning template applications](#)
You configure image enhancement at the page level on Learning template applications to remove lines, shading, misalignment, and other imperfections that can interfere with the recognition process.
 - [Locating fields in Learning template applications](#)
You configure recognition on the field level of Learning template applications to locate the fields that you want to capture and convert those fields into character-based data.
 - [Setting up field validation in Learning template applications](#)
You configure validation on the field level of Learning template applications to determine whether the captured data complies with the data integrity rules that are defined in your business requirements. Validation has to be configured in Datacap Studio.

Parent topic: [Getting started on FastDoc](#)

Jobs available in Learning template applications

The following jobs are available in Learning template applications.

About this task

DemoSingleTIFFS

Imports single-page TIFF images from the file system using rules.

DemoWebScan

Use this job if you are scanning using Datacap Web Client and manually selecting files from the file system.

DemoNavigatorScan

Use this job if you are scanning using Datacap Navigator and manually selecting files from the file system.

DemoMultiFormat

Imports any supported file type from the file system using rules. This job is similar to DemoSingleTIFFS, but allows for all supported file types in addition to TIFF images. All files are converted to single-page TIFF files as part of processing. Supported file types include:

- Multipage TIFF images
- Bitmap images
- ZIP archives
- PDF
- Microsoft Word, Outlook, and Excel documents
- HTML
- Text files (RTF or TXT)

FlexIDSingleTIFFs, FlexIDWebScan, FlexIDNavigatorScan, and FlexIDMultiFormat

These three jobs are similar to DemoSingleTIFFs, DemoWebScan, DemoNavigatorScan, and DemoMultiFormat respectively, but they also include a task to manually identify the pages in your documents and assign page types.

Parent topic: [Learning template application configuration](#)

Related tasks:

[Configuring export options](#)

Setting up fields on Learning template applications

The Learning template uses generic document types, page types, and field types that are provided with the template. You can add other files types that you want to capture with your application.

Procedure

To set up documents on Learning template applications:

1. Start FastDoc in Datacap Server mode.
2. Log in to your application created from the Learning template.
3. Click the Configure documents, pages, and fields icon.
4. Select the Main_Page and click Add Field.
5. Select Create New to create a new field type and enter a field name. For example: Last Name.
6. Select Use Existing if you want to use a field type that is already defined.
7. Click Add.
8. Add the remaining fields that you want to include on the page.
9. Click Settings and configure the field settings by using the hover help text for each field as a guide.
10. Click Save.

Parent topic: [Learning template application configuration](#)

Related tasks:

[Configuring rulesets for the application](#)

[Testing rulesets](#)

Setting up image enhancement on Learning template applications

You configure image enhancement at the page level on Learning template applications to remove lines, shading, misalignment, and other imperfections that can interfere with the recognition process.

Procedure

To set up image enhancement on Learning template applications:

1. Start FastDoc in Datacap Server mode.
2. Log in to your application created from the Learning template.
3. Click the Configure documents, pages, and fields icon.
4. Select the Other page in the Batch Structure pane and click the Ruleset tab.
5. Select the Image Enhancement ruleset.
6. Click Open an image file and select the image.
7. Select the enhancement options that you want to run to clean up the image. These options are run when you process the image at run time.

8. Click Test to test the ruleset on a temporary batch, and save the ruleset by clicking Save.
9. Run these steps on every page for which you want to run image enhancement.

Parent topic: [Learning template application configuration](#)

Locating fields in Learning template applications

You configure recognition on the field level of Learning template applications to locate the fields that you want to capture and convert those fields into character-based data.

Procedure

To set up field recognition on Learning template applications:

1. Start FastDoc in Datacap Server mode.
2. Log in to your application created from the Learning template.
3. Click the Configure documents, pages, and fields icon.
4. Select the field in the Batch Structure pane and click the Ruleset tab.
5. Select the Recognize Pages and Fields ruleset.
6. Check the Read Field checkbox. Select the options that you want to run by using the hover text help as a guide. These options are run when you process the image at run time. Options are provided for reading machine text, hand print, barcodes, or checkboxes.
7. Run these steps on every field for which you want to run recognition.

Parent topic: [Learning template application configuration](#)

Setting up field validation in Learning template applications

You configure validation on the field level of Learning template applications to determine whether the captured data complies with the data integrity rules that are defined in your business requirements. Validation has to be configured in Datacap Studio.

Procedure

To set up field validation in Learning template applications:

1. Start Datacap Studio and log in to the application you created using the Learning template.
2. On the Document hierarchy tab, select the field that you want to configure for validation.
3. On the Rulesets tab, select the Validate Fields ruleset.
4. On the Properties tab, click Settings.
5. Check Validate this field.
6. Configure the validation options that you want to run by using the hover text help as a guide. These options are run when you process the image at run time.
7. Run these steps on every field for which you want to run validation.

Parent topic: [Learning template application configuration](#)

Configuring export options

The Export task in the workflow for your job contains an Export ruleset that exports documents to a file system. You can add rulesets to export documents to the IBM® Content Manager, IBM FileNet® Content Manager, and Box repositories for storage.

About this task

You can add the Export to IBM Content Manager, Export to IBM FileNet Content Manager, or Export to Box ruleset to the Export task in your job. Then, you configure these rulesets on the batch, document, and field levels of the batch structure.

Procedure

To configure rulesets to export documents:

1. Start FastDoc in Datacap Server mode.
2. Click the Configure workflow icon.
3. Add the wanted Export ruleset to the Export task.
 - a. In the Jobs pane, select the job for which you want to add a ruleset.
 - b. In the Rulesets pane, select the ruleset and drag it to the header of the Export task in the job.
 - c. Right-click and select Remove to delete the existing Export ruleset from the task.
 - d. Click Save.
4. Configure the Export ruleset at the batch, document, or field level.

Important: Select each document and field that should be exported.

 - a. In the Configure Documents and Pages pane, select the batch structure level at which you want to run the ruleset. You configure the connection settings to the repository at the batch level.
 - b. Click the Ruleset tab and select the Export ruleset to display its settings. The export settings that apply to the selected batch structure level are displayed.
 - c. Configure the ruleset settings by using the hover help text as a guide.
 - d. Click Save.

- [Format of the exported data file](#)

You can use FastDoc to export documents. FastDoc exports the document image file and its associated data file to the folder that you specified. These files share a name, but have different file extensions.

- [Confirming a successful export](#)

After indexing and verification tasks are completed for a batch, FastDoc exports the images and index data. The exported information goes to the location that is specified in the Export ruleset.

Parent topic: [Getting started on FastDoc](#)

Related tasks:

[Jobs available in Forms template applications](#)

[Jobs available in Learning template applications](#)

Format of the exported data file

You can use FastDoc to export documents. FastDoc exports the document image file and its associated data file to the folder that you specified. These files share a name, but have different file extensions.

FastDoc supports two export formats for the data file: Character Delimited Value and XML. The exported data file defaults to Character Delimited Value with a file extension of CSV.

For both formats, you can include or exclude the following information from the export file.

- Document type
- Full path to the document image file
- File name of the document image file

Parent topic: [Configuring export options](#)

Confirming a successful export

After indexing and verification tasks are completed for a batch, FastDoc exports the images and index data. The exported information goes to the location that is specified in the Export ruleset.

About this task

If there were problems during the export, a warning message is generated. Details about the problem can be found in the log files in the C:\Datacap\FastDoc\batches\

-
- For export to IBM® FileNet® Content Manager, go to the appropriate Object Store to view the images and index entries.
- For export to IBM Content Manager, go to the specified folder on Content Manager to view the images and index entries.
- For export to a network folder, go to the appropriate location on your network. View the images and text or XML files that contain the index entries.

Parent topic: [Configuring export options](#)

Application configuration on FastDoc

You can configure your application for FastDoc by setting up batch profiles and adding rulesets to the FastDoc workflow.

Configuring your application for FastDoc also involves document planning, scanner setup, ruleset configuration, and index field definition.

- [Preparing for document scanning and indexing](#)
If you know which types of documents you want to process, you can accurately capture the most useful index data from those documents. These optional steps help you plan your document scanning, indexing, and storage.
- [Setting up a scanner for FastDoc](#)
You can attach a scanner to your FastDoc computer and scan your paper documents. FastDoc can process documents that are scanned on TWAIN or ISIS scanners, and image files of documents that were previously scanned or faxed.
- [Setting up batch profiles in Local mode](#)
You can create custom batch profiles on FastDoc in Local mode to configure the workflow tasks for your jobs.
- [Setting up batch profiles in Datacap Server mode](#)
Batch profiles are not needed in Datacap Server mode because you configure these tasks in Datacap Web Client when FastDoc is connected to the Datacap Server.
- [Adding rulesets to a FastDoc workflow](#)
You can add rulesets to the tasks in the workflow and run them on the applicable levels of the batch structure.
- [Configuring rulesets for the application](#)
After you set up the documents and fields to process, you can configure the rulesets to run on the application that is based on the template that you are using.
- [Testing rulesets](#)
You can test the rulesets that you applied to your documents, pages, and fields and make the appropriate changes before you process batches.
- [Defining an index field with keywords](#)
Keywords can locate an index value by using a field label that might vary when the index value is always

in the same location relative to the field label. You can define multiple keywords for each index field. You can also define each keyword as either a string or a regular expression.

- [Defining index field validation by using a database](#)
You can validate a captured index value by using a column in an external database. You can also populate an index field by using a value that is retrieved from a database.
- [Using Click N Key to capture data](#)
Use the Click N Key feature to select the index data in the Verify pane that you want to capture.

Parent topic: [Preparing and running FastDoc applications](#)

Preparing for document scanning and indexing

If you know which types of documents you want to process, you can accurately capture the most useful index data from those documents. These optional steps help you plan your document scanning, indexing, and storage.

Procedure

To prepare for document scanning and indexing:

1. Identify and gather samples of the documents that you want to scan, index, and store by using FastDoc. You can use the Forms Template application for documents that you created and your customers send back to you, like contract paperwork for your services or tax forms. You can use the Learning Template for documents that are created by other users. For example, invoices from different vendors for items that you purchased, confirmations of shipments, or car rental contracts.
2. Group these documents into types and assign a unique *Document Type* name to each group.
3. For each group of documents or *Document Type*:
 - a. Look at the variations and identify the types of index data in the index fields that you want to capture for each group.
 - b. Identify the name that you want the Operator to see for each field.
 - c. Determine the sequence in which you want the fields to display on the Verify pane during indexing. This sequence is also the sequence in which the values appear in the exported data file.
 - d. Decide whether you want to export the index values to a data file. You use a character delimited value or XML format.
4. For each Index field:
 - a. Identify whether you want the data to be located by using a zone. For each unique image, the data is always in the same general location. The customer address is in the same place on your order form. If the data is located by using a list of keywords that match the field label on the image, the invoice number can be located by using the string or `INVOICE NUMBERInvoice#`
 - b. Note the data characteristics of the values that you are capturing. For example, letters, numbers, contains punctuation, date, minimum and maximum number of characters.
 - c. Identify whether you want to validate the data that you are capturing. If so, identify the types of validations you want to do for each index field. Such as required/optional, value must exist in an external database, data characteristics must match.
 - d. Identify whether you want to use data in an external database to populate more index fields. For example, use the customer number from the order form to retrieve the postal code of the customer.
5. Identify the format (TIFF or PDF) to use for the exported documents and where you want to store the output. When you export documents to the following repositories or file.
 - IBM® FileNet® Content Manager: you need the URL, user ID, and Password of the Content Manager site and the Object Store into which you want to upload the documents. You must know the Document Class to which to map the exported document.

- IBM Content Manager: you need the server name, user ID, and Password of the IBM Content Manager site and the name of the folder into which you want to upload the documents.
- File: determine the network folder, local drive, mapped drive, or UNC path information and the folder name you want to use. The Windows user ID of the FastDoc Operator or Administrator must have network permissions to write to this target folder.
- Box: For information about Box export configuration, see [Configuring Box Connector actions](#).

Parent topic: [Application configuration on FastDoc](#)

Related tasks:

[Setting up a scanner for FastDoc](#)

Setting up a scanner for FastDoc

You can attach a scanner to your FastDoc computer and scan your paper documents. FastDoc can process documents that are scanned on TWAIN or ISIS scanners, and image files of documents that were previously scanned or faxed.

About this task

To get started without attaching a scanner, you can use the sample pre-scanned images that are provided with FastDoc.

FastDoc uses the top third of the first page of each document to identify that document. The scanner settings must match the actual size of the paper that is being scanned. For example, when you scan letter-sized pages, ensure that the scanner is set up for letter-sized paper and not legal-sized paper.

Procedure

To set up a scanner for FastDoc:

1. Follow the instructions of the scanner manufacturer to attach the scanner, install, and configure the scanner software and driver software.
2. Make sure you can scan successfully by using the scanner and any TWAIN or ISIS driver-based image capture software product. If you cannot scan documents outside of FastDoc, you cannot scan documents when you are using FastDoc.
3. Set your scanning resolution setting that is based on your unit of measure:
 - For inches, a scanning resolution of 200 is suggested
 - For centimeters, a scanning resolution of 78 is suggested

Tip:

When you start scanning documents, it is best to use the same scanning resolution for all documents. Do not change the resolution unless you must to improve your recognition results.

4. Start FastDoc.

Parent topic: [Application configuration on FastDoc](#)

Related tasks:

[Creating a batch by using a scanner](#)

Setting up batch profiles in Local mode

You can create custom batch profiles on FastDoc in Local mode to configure the workflow tasks for your jobs.

About this task

When you create a batch profile, you specify the profile name, type, and description. You also format rules for the folders, documents, and pages in the batch. You then create tasks for the batch profile and map them to the application, workflow, and task profile where you want to run them. FastDoc provides a batch profile that you can use as an example when you create your customized batch profiles.

The Administrator creates the batch profile and configures the tasks to run for the profile. Then, you can select the batch profile by name on the main Datacap window.

Procedure

To set up a batch profile in Local mode:

1. Start FastDoc in Local mode.
2. Click Configure local profile at the Datacap window.
3. Click Add to display the New Profile tab on the Batch Profiles window.
4. In the Batch Profile Name field, type the name of the batch profile. The batch profile name is displayed on a New Profile tab.
5. In the Type and Description fields, enter more information to further identify the batch profile.
6. Use the following information in format ID fields to create batch folders, documents, and pages.
 - o Batch ID Format: the year, month, day, hour, minute, result size, and computer name
 - o Document ID Format: the batch ID and document number
 - o Page ID Format: the page number within the document
7. Create a Scan task for the batch profile.
 - a. Click New Task to create tasks for the batch profile.
 - b. Type the name of the Scan task in the Task Name field. For example, you might use a name like Scan Tax Forms.
 - c. In the Task Options section, select a scan option from the list, such as Disk Scan, ISIS Scan, or TWAIN Scan.
 - d. In the Application field, select the application to run the task. The Workflow field is filled with the application name.
 - e. In the Task Profile field, select the ruleset to run on the task. For example, for a Disk Scan you can run Scan From Disk <Single TIFFS> or Scan From Disk <Multi Format>.
8. Create a PageID task for the batch profile. This task is used by the Forms template only.
 - a. Click + to create another task for the batch profile.
 - b. Type the name of the PageID task in the Task Name field.
 - c. In the Task Options section, select Rulerunner. The application name is displayed in the Application and Workflow fields.
 - d. In the Task Profile field, select PageID.
9. Create a Profiler task for the batch profile.
 - a. Click + to create another task for the batch profile.
 - b. Type the name of the Profiler task in the Task Name field.
 - c. In the Task Options section, select Rulerunner. The application name is displayed in the Application and Workflow fields.
 - d. In the Task Profile field, select Profiler.
10. Create a Verify task for the batch profile.
 - a. Click + to create another task for the batch profile.
 - b. Type the name of the Verify task in the Task Name field.
 - c. In the Task Options section, select Verify. The application name is displayed in the Application and Workflow fields.
 - d. In the Task Profile field, select Verify.
11. Create an Export task for the batch profile.

- a. Click + to create another task for the batch profile.
 - b. Type the name of the Export task in the Task Name field.
 - c. In the Task Options section, select Rulerunner. The application name is displayed in the Application and Workflow fields.
 - d. In the Task Profile field, select Export.
12. Click Save to save the batch profile and its tasks. You run the batch profile from the Batch Profile window.
 13. Click Remove to delete the batch profile. The tab for the batch profile is removed from the Batch Profile and the Select a Batch windows.

Parent topic: [Application configuration on FastDoc](#)

Related tasks:

[Creating a batch with pre-scanned images](#)

[Creating a batch by using a scanner](#)

Setting up batch profiles in Datacap Server mode

Batch profiles are not needed in Datacap Server mode because you configure these tasks in Datacap Web Client when FastDoc is connected to the Datacap Server.

About this task

You use the Datacap Web Client Administrator or Datacap Navigator to configure your applications to run on the FastDoc client in Datacap Server mode.

Procedure

To set up batch profiles in Datacap Server mode:

1. Start FastDoc in Datacap Server mode.
2. Open Internet Explorer, and enter the IP address of the Datacap Web Client or Datacap Navigator followed by the alias for the website. For example, enter `http://127.0.0.7/tmweb.net` or `http://WebServerName/tmweb.net`.
3. Click the Administrator tab and select Workflow.
4. Click the name of your application to expand the tree.
5. Select the Job for which you want to add a workflow tasks and click New.
6. Enter a name for the task. For example, enter `Scan`, `Profiler`, `Verify`, or `Export`.
7. Click Create Setup and specify the values for the task.
8. Click OK.
9. Repeat these steps for all of the tasks that you want to add.

Parent topic: [Application configuration on FastDoc](#)

Related tasks:

[Creating a batch with pre-scanned images](#)

[Creating a batch by using a scanner](#)

Adding rulesets to a FastDoc workflow

You can add rulesets to the tasks in the workflow and run them on the applicable levels of the batch structure.

About this task

FastDoc does not automatically add a reference to a ruleset in the \rrs\collection.xml file. If a reference is needed, it must be done outside of FastDoc. FastDoc loads rulesets from the \rrs folder whether they are in the \rrs\collection.xml file or not.

When a ruleset that is not in \rrs\collection.xml or \rules\collection.xml is added to a task in FastDoc, the *.rul (.dll) and *.rul.config files are copied to the \rules folder. A reference is added to the \rules\collection.xml file.

First, you add the rulesets to the tasks in your workflow. Then, you configure the rulesets to run on the tasks at the batch, document, page, or field level of your batch structure.

Procedure

To add rulesets to a FastDoc workflow:

1. Start FastDoc in Datacap Server mode.
2. At the Datacap FastDoc screen, click the Configure workflow icon.
3. Add the ruleset to a task:
 - a. In the Jobs pane, select the job for which you want to run a ruleset.
 - b. In the Rulesets pane, select the ruleset and drag it to the header of the task in the workflow. For example, select Export to IBM Content Manager and drag it to the Export header.
 - c. Right-click and select Remove to delete a ruleset from a task.
4. Configure the batch structure item to run the ruleset:
 - a. Double-click the ruleset in the workflow task to display the ruleset settings in the Configure Documents and Pages window. Some rulesets cannot be configured in FastDoc. If the ruleset settings screen does not display, you must configure the ruleset in Datacap Studio.
 - b. Select the batch structure level at which you want to run the ruleset. For example, select Document to run the ruleset at the document level. A message displays if the ruleset cannot be run at the selected batch structure level. This message also indicates at which level this ruleset can be run.
 - c. Configure ruleset settings for the batch structure item. Hover help text is provided in each field to assist you.
 - d. Click Save.

Parent topic: [Application configuration on FastDoc](#)

Related tasks:

[Configuring rulesets for the application](#)

[Testing rulesets](#)

Configuring rulesets for the application

After you set up the documents and fields to process, you can configure the rulesets to run on the application that is based on the template that you are using.

About this task

You configure rulesets to run on the batches, documents, pages, and fields that are in the application in Datacap Server mode.

Procedure

To configure rulesets on documents:

1. Start FastDoc in Datacap Server mode.
2. On the Datacap window, click the Configure workflow icon.

3. Select the job that you want to run and click the Configure documents, pages, and fields icon.
4. In the Batch Structure pane, select the name of the application and click the Rulesets tab.
5. Configure the following rulesets for the appropriate batch structure level in the application:

Table 1. Rulesets to configure for the Forms and Learning applications

App licat ion	Batch Structur e Level	Ruleset
For ms	Batch	Identify Pages to identify non-classified pages in the document.
		Convert Images to convert the electronic images to a TIFF file format.
		Create Documents to arrange page files into documents.
	Docume nt	There are no ruleset settings for this level.
	Page	Image Enhancement to clean up imperfections in the scanned image.
		Recognize Pages and Fields to run recognition steps task on pages.
	Field	Recognize Pages and Fields to run recognition steps on fields.
		Validate Fields to determine whether the captured data conforms to the specified business rules.
Lear ning	Batch	Populate Fields By Using Keywords to define keywords that can locate the index value by using a field label.
		Convert Images to convert the electronic images to a TIFF file format.
		Export to FileNet Content Manager to configure the connection to the FileNet® Content Manager repository. You can configure the application for any one of the supported Export rulesets.
	Docume nt	Populate Fields By Using Keywords to define keywords that can locate the index value by using a field label.
		Export to FileNet Content Manager to configure the connection to the FileNet Content Manager repository. You can configure the application for any one of the supported Export rulesets.
	Page	Image Enhancement to clean up imperfections in the scanned image.
	Field	Populate Fields By Using Keywords to define keywords that can locate the index value by using a field label.
		Export to FileNet Content Manager to configure the connection to the FileNet Content Manager repository. You can configure the application for any one of the supported Export rulesets.

6. Use the Test tab to confirm that the processing is working correctly and make corrections as needed.
7. Start processing real documents.

Parent topic: [Application configuration on FastDoc](#)

Related tasks:

[Adding rulesets to a FastDoc workflow](#)

[Testing rulesets](#)

Testing rulesets

You can test the rulesets that you applied to your documents, pages, and fields and make the appropriate changes before you process batches.

About this task

You use this feature to test your rulesets by using a temporary batch.

Procedure

To test rulesets:

1. Start FastDoc in Datacap Server mode.
2. In the Batch Structure tree, select the batch item that you want to configure and click the Ruleset tab.
3. Select the ruleset, configure it for the batch item, and click Save.
4. In the Test pane, click Add File. You must add a file to test. If you do not add a file, FastDoc cannot test most of the rulesets. However, some rulesets might be tested anyway.
 - a. Go to the C:\Datacap*appname*\images\Input folder and select the first page that you want to test.
 - b. Continue to add the pages that you want to test one page at a time. The files are put into a temporary folder for the batch. A Page ID and Page Type is assigned to each of the pages.
 - c. Click the Arrow buttons to select the page that you want to test. The image of the selected page is displayed next to the Test pane.
5. Select a task profile and click Test to test all of the rulesets in the task profile. You do not have to test all of the ruleset in the task profile. You can run the rulesets in sequence to get to the ruleset that you want to test. Click the rulesets that run after the ruleset that you want to test to remove them from the test.
 - a. Look at the Status field for the test results. Fail means that the validations that were run on some of the fields failed.
 - b. Click each of the fields in the Batch Structure tree to find the fields that failed validation. The validation error is displayed in the Message field.
 - c. Enter the correct field information in the Value field.
 - d. Select the task profile and click Test to retest the corrected rulesets until the Status field displays Pass.
6. Click Reset to delete the files from the temporary batch folder and start over.

Parent topic: [Application configuration on FastDoc](#)

Related tasks:

[Adding rulesets to a FastDoc workflow](#)

[Configuring rulesets for the application](#)

[Setting up documents on Forms template applications](#)

[Setting up fields on Learning template applications](#)

Defining an index field with keywords

Keywords can locate an index value by using a field label that might vary when the index value is always in the same location relative to the field label. You can define multiple keywords for each index field. You can also define each keyword as either a string or a regular expression.

About this task

When you define both a zone and keywords for a field, FastDoc attempts to locate the index value by using the zone first. If the value is found by using the zone, the keyword criteria is ignored. If no value is found by using the zone, FastDoc then attempts to locate the index value by using the list of keywords.

Procedure

To define an index field with keywords:

1. Start FastDoc in Datacap Server mode.
2. On the Datacap window, click the Configure documents, pages, and fields icon.
3. In the Batch Structure pane, select the field that you want to define with keywords.
4. Click the Ruleset tab and select Populate Fields Using Keywords.
5. Click Add Term and enter the keyword or regular expression to use to define the field.
6. Select whether to locate the first occurrence of the keyword or the last occurrence of the keyword.
7. Click Add Locate Movement and select the direction in which to search for the keyword.
8. Specify how to group the words that you want to use in the found file.
9. Click Save.

Parent topic: [Application configuration on FastDoc](#)

Related tasks:

[Indexing and verifying batches](#)

Defining index field validation by using a database

You can validate a captured index value by using a column in an external database. You can also populate an index field by using a value that is retrieved from a database.

Procedure

To define index field validation or population by using a database:

1. Start FastDoc in Datacap Server mode.
2. On the Datacap window, click the Configure documents, pages, and fields icon.
3. In the Batch Structure pane, select the field that you want to validate by using a database.
4. Click the Ruleset tab and select the Validate Fields ruleset.
5. Select the check box for the Lookup section.
6. In the Database field, enter a connection string that can connect FastDoc to your external database. Based on the type of database and the driver that you are using, you might also need to set up a System DSN.
7. Enter an SQL statement that determines whether only validation is done, or whether a value from a different database column is returned.
8. Enter a comma and True after the SQL statement to populate the index field with the result.
9. Click the Populate Field check box.
10. Click Save.

Parent topic: [Application configuration on FastDoc](#)

Related tasks:

[Indexing and verifying batches](#)

Using Click N Key to capture data

Use the Click N Key feature to select the index data in the Verify pane that you want to capture.

About this task

There are two ways to capture data from the image.

Procedure

To use Click N Key to capture data:

1. Click in the Index field in the Verify pane.
2. Perform one of the following steps:
 - o Click the data on the image. The selected data displays in the Index field.
 - o Click and drag the mouse on the image to outline the data you want to capture. Release the mouse to make the zone visible on the image. If you drew the zone around recognized data, the data displays in the Index field.
3. For the second option, if the data does not display in the Index field, repeat the step but outline a larger area around the data.
4. If the data still does not display, the Perform Full Page Recognition setting is turned off for this batch.
 - a. Put the batch on hold and delete the batch.
 - b. Reprocess the batch, click Scan Settings, and ensure that Perform Full Page Recognition is enabled.
5. Repeat the capture data from image steps.

Parent topic: [Application configuration on FastDoc](#)

Related tasks:

[Indexing and verifying batches](#)

Document processing on FastDoc

FastDoc automates the capture of index data from computer or manually printed documents, which eliminates manual data entry. You can create batches of documents, run indexing and verification tasks, and export the index data and document content to a storage location.

- [Preparing paper documents for scanning](#)
FastDoc requires little preparation for paper documents, but processing documents runs smoother if you check a few simple things.
- [Running FastDoc in Local mode](#)
In Local Mode, FastDoc runs as a stand-alone client that scans, manually indexes, and uploads batches without using Datacap Server/Datacap server.
- [Running FastDoc in Datacap Server mode](#)
You can run the FastDoc as a client to Datacap Server to scan, index, verify, and upload documents to Datacap Server. In Datacap Server mode, you can use Datacap Studio and Application Manager for more application development.
- [Displaying a page or document](#)
You can manage the pages in an FastDoc document by using the icons in the Control Bar.
- [Deleting pages](#)
You can delete pages from your FastDoc batch from the Image Stream pane.

Parent topic: [Preparing and running FastDoc applications](#)

Preparing paper documents for scanning

FastDoc requires little preparation for paper documents, but processing documents runs smoother if you check a few simple things.

Procedure

To prepare paper documents for processing:

1. Ensure that all of the documents in the batch can be separated the same way:
 - a. Insert patch code separator pages for all documents in a batch when you want FastDoc to automatically separate documents for you at the separator pages.
 - b. Print and use copies of the Separator.PDF file that is provided in the C:\Datacap\
<appname>\Images\Input folder. The FastDoc patch code separator page is not exported when the document is exported.
 - c. Use the Pages per Document field to automatically separate documents when all of the documents have the same number of pages.
 - d. Manually identify the first page of each document in the batch.
2. Ensure that you have all pages of all documents in the correct page number sequence.
3. Ensure that your pages do not jam in the scanner, smooth all pages, and remove all staples and paper clips.

Parent topic: [Document processing on FastDoc](#)

Running FastDoc in Local mode

In Local Mode, FastDoc runs as a stand-alone client that scans, manually indexes, and uploads batches without using Datacap ServerDatacap server.

About this task

In Local mode, you can configure applications by creating custom profiles to run tasks for batches of documents and images.

If you are using a scanner, start your scanner before you log in to FastDoc.

Procedure

To run FastDoc in Local mode:

1. Start FastDoc, select Local and click Login.
 2. Click Send Forms to Datacap to run create batch profiles in Local mode and upload them to Datacap Server in the background.
 3. In the Add Batch field, select the application for which you want to create a new batch.
 4. In the Existing Batch field, double-click the batch that you want access.
 5. Select the batches that you completed and click Purge Finished Batches to remove them.
 6. Select the batches that you no longer want to process and click Delete Selected Batches to remove them.
 7. If you try to exit after you start FastDoc, your batch is put on hold before exiting. If FastDoc is processing a batch when you try to exit, FastDoc completes this processing before you can exit.
- [Opening an existing batch](#)
You can put batch processing on hold when you do not have the time or the information to complete it. When you are ready to continue, you can open the batch that you put on hold and work on it.
 - [Creating a batch with pre-scanned images](#)
You can run the FastDoc Disk Scan option to create a batch of image files that are stored in a location that is accessible from your computer.

- [Creating a batch by using a scanner](#)
You can run the FastDoc TWAIN Scan or ISIS Scan option to create a batch of documents from the image files that you scanned.
- [Indexing and verifying batches](#)
After the FastDoc Recognition task is run, review the results in the Verify pane and make corrections where needed.

Parent topic: [Document processing on FastDoc](#)

Opening an existing batch

You can put batch processing on hold when you do not have the time or the information to complete it. When you are ready to continue, you can open the batch that you put on hold and work on it.

Procedure

To open an existing batch:

1. Start FastDoc in Local mode. The Datacap window opens with the count of Active Batches and Finished Batches.
2. If the batch you want to open is displayed, double-click the batch to open it.
3. If the batch you want to open is not displayed, click Ellipsis until the batch appears, then double-click the batch to open it.
4. If you are an Operator, you can do the following steps.
 - a. At the Scan pane, create a batch by using a scanner or pre-scanned images.
 - b. At the Verify pane, if you want to complete indexing and verification of the batch, continue with the appropriate step in the indexing and verification process.
5. If you are an Administrator, you can do the following steps.
 - a. At the Scan pane, create a batch by using a scanner or pre-scanned images.
 - b. Open the Verify pane and set up a document type and its index fields.

Parent topic: [Running FastDoc in Local mode](#)

Related tasks:

- [Setting up batch profiles in Local mode](#)
- [Setting up batch profiles in Datacap Server mode](#)
- [Creating a batch with pre-scanned images](#)
- [Creating a batch by using a scanner](#)

Creating a batch with pre-scanned images

You can run the FastDoc Disk Scan option to create a batch of image files that are stored in a location that is accessible from your computer.

About this task

FastDoc can process pre-scanned image files with the following file extensions:

- bmp
- Jpg, jpeg
- Pdf (single and multi-page files)
- png
- Tif, tiff (single and multi-page files)

Procedure

To create a new batch with pre-scanned images:

1. Start FastDoc in Local mode.
2. Click the Batch Profile that you want to run. The FastDoc main window displays the Scan pane.
3. If the Disk Scan pane is not displayed, click View and select Disk Scan.
4. Click Scan. The Select the images to add to the batch window opens.
5. Go to the folder that contains the images you want to process and select the images to process. The default location for the sample images is C:\Datacap\appname\Images\Input.
6. Click Open. The images are downloaded into FastDoc and thumbnails of each image are displayed in the Image Stream pane.
7. Repeat the scanning procedure to include more images in this batch. Ensure you downloaded all of the pages of each multi-page document. You can use the Control Bar to display a specific page or document to delete one or more images, and sequence pages.
8. To put the batch on hold, click ▲ next to Submit and select Hold.
9. To cancel processing the batch, click ▲ next to Submit and select Cancel.
10. Click Submit to run recognition processing.
11. Click ✓ to validate this document and go to the next document and make any needed corrections. After the Validation task is completed, Rulerunner runs the Export ruleset that you configured for this Batch Profile.

Parent topic: [Running FastDoc in Local mode](#)

Related tasks:

[Indexing and verifying batches](#)

Creating a batch by using a scanner

You can run the FastDoc TWAIN Scan or ISIS Scan option to create a batch of documents from the image files that you scanned.

About this task

The instructions for using a scanner to download documents into FastDoc are different from using pre-scanned images.

Procedure

To create a batch by using a scanner:

1. Prepare your paper documents for scanning.
2. Ensure that your scanner is started.
3. Start FastDoc in Local mode.
4. Click the Batch Profile that you want to run. The FastDoc main window displays the Scan pane.
5. Click View and select TWAIN Scan, or ISIS Scan. The pane for the selected driver is displayed.
6. Click Source to display a list of available scanners. If your scanner does not appear, the driver of the scanner was not installed properly.
 - a. Close FastDoc and follow the instructions of the scanner manufacturer to install or reinstall the driver of the scanner.
 - b. Restart FastDoc.
 - c. Return to Source and select the driver.
7. Click Options for TWAIN Scan or Advanced for ISIS Scan and select the appropriate settings.
 - a. Click Duplex to scan both sides of each page.

- b. Click Landscape when your scanner is feeding paper in landscape orientation.
- c. Select the appropriate image format options for the Mode that you selected.

Attention:

For Kodak scanners, if the two-sided option is selected on the Kodak scanner, your images are scanned in duplex mode regardless of the Duplex setting on FastDoc. If the one-sided scan option is selected on the scanner, you can set duplex mode. Click Duplex on FastDoc or do nothing and scan the images in one-sided mode.

8. Click OK.
9. Ensure that the Resolution setting is set to the scanning resolution (200 dpi or 78 DPCM). The current setting is displayed beneath the dial. Click above the dial, and move your cursor horizontally over the dial to change the setting.
10. Change the settings for brightness and contrast you do not want the default settings. Click above the dial, and move your cursor horizontally over the top of the dial to change the setting.
11. Load your pages into the scanner and click Scan. Messages display as scanning begins. The scanned image of the first page is displayed in the Active Image pane. Thumbnails of each of the scanned pages appear in the Image Stream pane. You can use the Control Bar to display a specific page or document to delete one or more images, and sequence pages.
12. To put the batch on hold, click ▲ next to Submit and select Hold.
13. To cancel the batch, click ▲ next to Submit and select Cancel.
14. Click Submit to run recognition processing.
15. Click ✓ to validate this document and go to the next document and make any needed corrections. After the Validation task is completed, Rulerunner runs the Export ruleset that you configured for this Batch Profile.

Parent topic: [Running FastDoc in Local mode](#)

Related tasks:

[Preparing paper documents for scanning](#)
[Indexing and verifying batches](#)

Indexing and verifying batches

After the FastDoc Recognition task is run, review the results in the Verify pane and make corrections where needed.

About this task

When FastDoc completes the Recognize task, the FastDoc window is displayed with the Verify pane, the Active Image pane, and the Control Bar.

Each Verify pane looks and behaves differently depending on the fields and the types of validation that the administrator defined for the Document Type. The pane displays fields that might contain captured data. Fields with low confidence recognition are displayed in yellow. The fields that failed validation rules are displayed in pink.

Procedure

To index and verify your batch:

1. When the Document type selected by FastDoc is incorrect, you can change it.
2. When the low confidence data is correct, tab out of the field.
3. Make corrections to fields that contain incorrect low confidence data or failed validations by using any of these methods or tools:
 - You can include missing data or replace problem index values by using the Click N Key.

- You can also type in the correct values.
- 4. Display any image in the batch, change the image size, or change the batch by using any of these methods or tools:
 - a. Select the image that you want to display by clicking it in the Active Image pane.
 - b. Use the buttons on the Control Bar to display a specific page or document, or use the other buttons to delete one or more images, and sequence pages.
 - c. Use keyboard shortcuts to move focus among the various panes, panes, and manipulate the image.
- 5. Click Next Problem to rerun validations and display the next image that failed validations.
- 6. If an error message is displayed that indicates that a validation failed, the field value must be corrected before you can continue to the next document. Click OK, correct the error, and repeat the previous step.
- 7. At the end of the batch, if the document or page count you entered when you created the batch:
 - Does not match the actual number of documents or pages that are being processed, a warning is displayed. This issue usually happens because you deleted one or more documents or pages. Click Yes to continue.
 - Matches the number of batches that are being processed, a message is displayed indicating there are no more problems.
- 8. Click OK. FastDoc exports the images and index values for the batch and displays the Datacap window where you can create another new batch.

Parent topic: [Running FastDoc in Local mode](#)

Related tasks:

[Defining an index field with keywords](#)

[Defining index field validation by using a database](#)

[Using Click N Key to capture data](#)

Running FastDoc in Datacap Server mode

You can run the FastDoc as a client to Datacap Server to scan, index, verify, and upload documents to Datacap Server. In Datacap Server mode, you can use Datacap Studio and Application Manager for more application development.

About this task

In Datacap Server mode, you use the Datacap Web Client Administrator or Datacap Navigator to configure the tasks on your applications that are run on the FastDoc client.

Procedure

To run FastDoc in Datacap mode:

1. Start FastDoc and select Datacap Server on the Welcome to Datacap window.
 - a. Select the Datacap application.
 - b. Type a valid user ID and Password for the workstation.
 - c. Type the ID of the workstation.
 - d. Click Login.
2. Click the task shortcuts for the selected application.
 - [Processing documents in Forms template applications](#)
Choose a job and process your documents through the tasks of your application workflow. This procedure processes images for the DemoMultiFormat job as an example.
 - [Processing documents in Learning template applications](#)
Choose a job and process your documents through the tasks of your application workflow. This

procedure processes images for the DemoMultiFormat job as an example.

Parent topic: [Document processing on FastDoc](#)

Processing documents in Forms template applications

Choose a job and process your documents through the tasks of your application workflow. This procedure processes images for the DemoMultiFormat job as an example.

Before you begin

Before starting this task, create your application by following the instructions in [Creating an application in Application Wizard](#) and selecting the Forms template.

The following steps describe how to process images for the DemoMultiFormat job in Forms template applications. For a list of the different types of jobs available in Forms template applications, see [Jobs available in Forms template applications](#).

Important: The DemoMultiFormat job will process documents found in the C:\Datacap\appname\images\Input_MultiFormat directory, so you need to copy all of the images and files that you want to capture to that directory before starting this task.

About this task

Rulesets that do not display an ellipse next to their name in the Workflow Editor must be configured in Datacap Studio by using actions.

Procedure

To process documents:

1. Start FastDoc in Datacap Server mode.
2. Log in to your application.
3. Click the VScan shortcut. This task runs a disk scan on the image files that are stored in the C:\Datacap\appname\images\Input_MultiFormat. Since you selected the DemoMultiFormat job for this task, the Import, Convert Images, and DeleteSourceImages rulesets are run. FastDoc then returns to the Shortcuts screen.
4. Click the PageID shortcut. This task enhances the image and identifies the page type. It then arranges the pages into documents based on the [Document Integrity](#) rules in the [Document Hierarchy](#) for your application, and assembles these documents in the batch. The PageID task runs in the background, and consists of rulesets that are run in the following order:
 - a. Image Enhancement: cleans up the imperfections in the image.
 - b. Identify Pages: identifies any non-classified pages. Each enabled identification technique at the batch level to run recognition on the image. The Fingerprinting recognition option is selected by default. If you select another option, you might have to redefine the zones on the page.
 - c. Create Documents: arranges the contents of a Page file into documents based on the [Document Integrity](#) rules in the [Document Hierarchy](#) for your application, and assembles these documents in the batch.
 - d. Document Integrity: checks the batch structure. If the page type or the document type is not recognized, the Routing step opens the Fixup Job.
5. Click the Fixup shortcut to resolve recognition issues with the page type ID or the document type.
 - a. Double-click the job that you want to fix. FastDoc displays the Verify screen for that job.
 - b. Click Document and select the document type to use.

- c. Click the inverted triangle next to Document and select the page type to use.
 - d. Click Submit.
 - e. FastDoc returns to the PageID task and reruns each page until every page is ready for the Profiler task profile.
Attention: FastDoc uses the Verify screen to run Fixup. You can use Datacap Desktop to run full Fixup functionality.
6. Click the Profiler shortcut. This task attempts recognize and validate the data in your documents. If the data is successfully validated, it is routed to the Export task. If there is low confidence that some data is valid, it is sent to be verified before it can be exported. The Profiler task runs in the background, and consists of rulesets that are run in the following order:
 - a. Recognize Pages and Fields: uses the recognition method that you specified, such as reading barcodes on a page, or reading text within page or field zones.
 - b. Validate Fields: sets up the conditions that must be met to validate the data in each field. The data in every field must meet these conditions.
 - c. Routing: marks the pages as passed, low confidence, depending on whether the data in the fields on the page were successfully validated. The forms that pass validation are routed directly to the Export task. Forms that don't pass are routed to a Verify Export job where they are corrected.
 7. Click the Export shortcut to run the ruleset that you configured for this job. The Export shortcut consists of rulesets that are run in the following order:
 - a. SetStatuses: sets the exporting status of the document.
 - b. PreExport: prepares the data for export to your repository.
 - c. Export: Writes the file to the specified location on the disk.
 - d. ProcessExceptions: handles exceptions to regular processing such as unwanted documents and rescans.

Parent topic: [Running FastDoc in Datacap Server mode](#)

Processing documents in Learning template applications

Choose a job and process your documents through the tasks of your application workflow. This procedure processes images for the DemoMultiFormat job as an example.

Before you begin

Before starting this task, create your application by following the instructions in [Creating an application in Application Wizard](#) and selecting the Learning template.

The following steps describe how to process images for the DemoMultiFormat job in Learning template applications. For a list of the different types of jobs available in Learning template applications, see [Jobs available in Learning template applications](#).

Important: The DemoMultiFormat job will process documents found in the C:\Datacap\appname\images\Input_MultiFormat directory, so you need to copy all of the images and files that you want to capture to that directory before starting this task.

About this task

Rulesets that do not display an ellipse next to their name in the Workflow Editor are configured in Datacap Studio by using actions.

Procedure

To process documents:

1. Start FastDoc in Datacap Server mode.
2. Log in to your application.
3. Click the VScan shortcut. This task runs a disk scan on the image files that are stored in the C:\Datacap\appname\images\Input_MultiFormat. Since you selected the DemoMultiFormat job for this task, the Import, Convert Images, and DeleteSourceImages rulesets are run. FastDoc then returns to the Shortcuts screen.
4. Click the Profiler shortcut. The shortcut will perform document separation, classification, recognition, data extraction, and flag field validation failures for the operator to review during verification. The Profiler task runs in the background, and consists of rulesets that are run in the following order:
 - a. Managed Rotation: enables rotation of the image.
 - b. Image Enhancement: cleans up the imperfections in the image.
 - c. PageID: identifies the path on which the document was captured and creates the document structure. If the path is the same as another document, the ruleset assumes that it is the same document type.
 - d. Create Documents: arranges the contents of a Page file into documents based on the [Document Integrity](#) rules in the [Document Hierarchy](#) for your application, and assembles these documents in the batch.
 - e. Recognize Pages and Fields: identifies the page types as Main_Page and Trailing_Page, inserts a Doc_Separator page in front of the Trailing_Pages, and inserts an Attachment_Separator page in front of the attachments. This job is not run on the attachments.
 - f. Fingerprint: builds a [fingerprint](#) with a fingerprint ID for each document type.
 - g. Locate: searches the fingerprint file for data (for example: a social security number, invoice number, or date).
 - h. Lookup: finds the [fingerprint class](#) for your fingerprint IDs.
 - i. Validate: validates the data in every field and makes corrections where needed.
 - j. Routing: routes the forms that passed validation to the Export task.
5. Click the Verify shortcut, which runs the Validate ruleset on the form to remember the field information and automatically process it the next time that it is recognized.
6. Click the Export shortcut to run the ruleset that you configured for this job. The Export shortcut consists of rulesets that are run in the following order:
 - a. SetStatuses: sets the exporting status of the document.
 - b. PreExport: prepares the data for export to your repository.
 - c. Export: Writes the file to the specified location on the disk.
 - d. ProcessExceptions: handles exceptions to regular processing such as unwanted documents and rescans.

Parent topic: [Running FastDoc in Datacap Server mode](#)

Displaying a page or document

You can manage the pages in an FastDoc document by using the icons in the Control Bar.

Procedure

To organize the pages in your document:

1. Click <Go to previous or Go to next>to move from page to page.
2. Click |<first or last>|to display the first or last page in the document.
3. Select the thumbnail of the page or document or right-click Page or Document and enter the number of the page or document you want to view.

Parent topic: [Document processing on FastDoc](#)

Deleting pages

You can delete pages from your FastDoc batch from the Image Stream pane.

About this task

If you delete all of the pages from a batch and click Next Problem, the message cannot continue without pages. Place batch on hold to delete it. is displayed. Click OK, then click Hold.

Procedure

1. Use the following procedures to delete pages from the batch:
2. Select the page that you want to delete and click X to delete the current page.
3. Select the first page that you want to delete and click X> to delete that page and all of the pages to the end of the batch.
4. Click <X> to delete all of the pages in the batch.

Parent topic: [Document processing on FastDoc](#)

Accounts Payable application

The Datacap Accounts Payable application uses optical character recognition and location rules to capture invoice data. It delivers the data to your accounts payable, ERP, Document Management, and other processing systems. You can run Datacap Accounts Payable in a thick client environment or on Datacap Web Client to automate the process of capturing invoices.

Datacap Accounts Payable is delivered with sample images and a multi-page invoice separator page for different pre-configured jobs. You can run the tasks in these jobs to use Datacap Accounts Payable out-of-the-box without having to configure more steps.

Datacap Accounts Payable includes the APT Add Demo Vendor utility program to be able to add vendors to a vendor list, which acts as a sample Accounts Payable Vendor database. The vendor list does vendor lookups and validations on your Accounts Payable Vendor database because Datacap Accounts Payable out-of-the-box cannot access the database.

- [Setting up Datacap Accounts Payable to process your invoices](#)
The installed Datacap Accounts Payable application cannot access your accounts payable or vendor database to do vendor lookups and validation. It also cannot access your purchase order database to match line items from your invoices to your purchase orders. You can simulate the vendor lookup and purchase order match steps to test the processing of your invoice images.
- [Datacap Accounts Payable client procedures](#)
Use the Datacap Accounts Payable client application to process the sample, pre-scanned invoices images that are included with the Datacap Accounts Payable product option.
- [Datacap Accounts Payable Web Client procedures](#)
You can use the Datacap Accounts Payable application to process pre-scanned images by running the default shortcuts on the Datacap Web Client.

Setting up Datacap Accounts Payable to process your invoices

The installed Datacap Accounts Payable application cannot access your accounts payable or vendor database to do vendor lookups and validation. It also cannot access your purchase order database to match line items

from your invoices to your purchase orders. You can simulate the vendor lookup and purchase order match steps to test the processing of your invoice images.

About this task

The VendorLookup database is intended for demonstrations and testing only. In a production environment, Datacap Accounts Payable must be connected to an ERP or other system to retrieve approved vendor names and vendor numbers.

Before you create your simulated database environment, you must shut down the Datacap software, remove the existing sample images, and replace them with your invoice images.

You add your vendors to the Demo Vendors list. Add your purchase order line items to the POTable in the database. Run Datacap Accounts Payable.

Procedure

Follow this procedure to set up Datacap Accounts Payable to process your invoices.

1. Shut down the Datacap components that are running on each computer in the following sequence.
 - a. Datacap client software for all applications, including Datacap Desktop, Datacap Web Client, Rulerunner, Datacap Studio, and so on.
 - b. Datacap Web Services and other web services, including Report Viewer, WRRS, Fingerprint Service, and so on.
 - c. Datacap Server Service
2. Replace the sample images with your invoice images:
 - a. Open Windows Explorer and go to the \\Server\Datacap\APT\Images\Input folder.
 - b. Make a backup copy of the folder.
 - c. Delete the sample images in the folder.
 - d. Copy your invoice images into the folder.

You can copy as many of the multi-page document separator sheet images as you need. Single page invoices that are placed at the start of the batch do not need separator sheets. But after a separator sheet is used in the batch, all of the remaining invoices in the batch must be separated by separator sheets.

- [Adding your vendors to the Demo vendors list](#)
To run customer invoices and pass validations, you need the vendor name, vendor postal code, and vendor number. You can use the Datacap Accounts Payable Add Demo Vendor tool to you add your vendor data to a simulated database. Then, you can process invoices from vendors other than the vendors in the APT samples.
- [Adding your purchase order line items to a simulated database](#)
You can simulate your purchase order database by adding entries to a purchase order table in the default Microsoft Access APTLook database. You add your purchase order line items to the simulated database for the Datacap Accounts Payable POLR feature. POLR matches the line items from your invoices to line items in purchase orders.

Parent topic: [Accounts Payable application](#)

Adding your vendors to the Demo vendors list

To run customer invoices and pass validations, you need the vendor name, vendor postal code, and vendor number. You can use the Datacap Accounts Payable Add Demo Vendor tool to you add your vendor data to a simulated database. Then, you can process invoices from vendors other than the vendors in the APT samples.

About this task

The Datacap Accounts Payable Add Demo Vendor is used for demonstration purposes only. In production systems, Datacap Accounts Payable gets the vendor data from the ERP system of the customer.

Procedure

To add your vendors to the Demo vendors list:

1. Open Windows Explorer, go to \\Server\Datacap\APT\dco_APT and double-click the Add Vendor Demo.exe file. The Add Demo Vendor window opens.
2. If information is displayed for a vendor in the vendor fields, click Clear.
3. For each of your vendors, enter a Vendor Name, a Vendor postal code. Then, enter a Vendor Number or select the Generate Random option and click Add. The vendor is added to the Demo vendors list.
4. Select a vendor name from the Demo vendors list and click Delete to remove that name from the list.
5. Close the Add Demo Vendor window when you added all of the vendors that are associated with your invoice images.

Parent topic: [Setting up Datacap Accounts Payable to process your invoices](#)

Adding your purchase order line items to a simulated database

You can simulate your purchase order database by adding entries to a purchase order table in the default Microsoft Access APTLook database. You add your purchase order line items to the simulated database for the Datacap Accounts Payable POLR feature. POLR matches the line items from your invoices to line items in purchase orders.

About this task

You can add the actual purchase orders that are related to one or more of your invoice images. Or you can use the invoice images to create the purchase order line item entries.

Procedure

To add your purchase order line items to the POTable in a simulated database:

1. Open Windows Explorer, go to \\Server\Datacap\APT\APTLook.mdb and make a copy of the database.
2. From the Windows Start menu, select your database client and select Microsoft Office Access. The Microsoft Access main window opens.
3. Select File > Open, go to \\Server\Datacap\APT\APTLook.mdb.
4. Right-click on the file and click Open.
5. Double-click POTable to open it. The POTable: Table is displayed with default data.
6. On the Edit menu, point to Go To and select Last. The row marker moves to the last row in the table.
7. On the Edit menu, point to Go To and select New Record. A row is added to the table.
8. For each line item, press Tab to move to the next column, and enter your data in each column. For example, enter the purchase order number, line item quantity, line item number, description of item, and purchase order line number.
9. Repeat the previous two steps to add the purchase order line items on which you want APT to do line item matching.
10. Select File > Save, then close the POTable window, the APTLook window, and the Microsoft Access window.

Parent topic: [Setting up Datacap Accounts Payable to process your invoices](#)

Datacap Accounts Payable client procedures

Use the Datacap Accounts Payable client application to process the sample, pre-scanned invoice images that are included with the Datacap Accounts Payable product option.

To run the Datacap Accounts Payable application, [start the Datacap Server Service](#) and run the APT tasks that process the images by using Datacap Desktop.

The sample images that are provided consist of single and multi-page invoices and a separator page. You can use the sample images to run Datacap Accounts Payable by using a virtual scan without having to connect a scanner to the Workstation.

The default shortcuts that are displayed when you start Datacap Accounts Payable that start and run various job-task combinations.

A job is a collection of tasks and rules that you run to process invoices. The pre-configured jobs on Datacap Accounts Payable include.

- Main Job
- Demo
- Demo-Dot Matrix
- Demo-FlexID
- Demo-Multipage TIFF
- Web Demo
- Web Demo-Dot Matrix
- Web Main
- Web Main-Dot Matrix

The Demo-FlexID job is used as the example in this documentation. The default shortcuts that display when you start Datacap Accounts Payable that start and run various job-task combinations.

The following table lists the default shortcuts that you run for the Demo-FlexID job and the sequence in which you run them on Datacap Accounts Payable.

Shortcut	Description
Disk Scan - APT	Selects the Demo-FlexID job and run the DiskScan task to bring the sample invoice images into Datacap Accounts Payable.
BatchProfiler - APT	Starts the Batch Profiler task that runs multiple image-processing functions. These functions include identification of pages, image cleaning such as removing speckles, image rotation, full page character recognition, and more. It also attempts to identify the invoice by comparing it to a library of known invoices.
Verify - APT	Starts the Verify task to display the sample invoice images and the data that was recognized in the Verify window. You can ensure that the data is complete and correct where needed. When an invoice is not a known invoice, the Verify window identifies the fields and data on it. You can save the fingerprint of the image so it is recognized the next time.

Shortcut	Description
Export - APT	Starts the Export task that completes image processing, saves invoice information, and exports field data to XML files. You can import these files into your Accounts Payable, Document imaging, or ERP system. Depending on the available batches and their current statuses, clicking the Background shortcut runs either the Batch Profiler task or the Export task.
Upload	Used by remote users who are using Datacap Web Client to upload batches to the Datacap Server.

- [Scanning invoice images on Datacap Desktop](#)
 You can run the APT VScan task to create a batch of invoices from the sample images that are provided with Datacap Accounts Payable. You can also set up an APT VScan task to scan images from a scanner or MFP device.
- [Preparing invoice images for processing on Datacap Desktop](#)
 You run the APT FlexID task on Datacap Desktop to prepare the invoice images for the processing that takes place in the Batch Profiler task. You can change the image size, orientation, sequence in the batch, and page type settings.
- [Processing invoice images on Datacap Desktop](#)
 After you scan the invoice images and prepare them for processing, use Datacap Desktop to run the APT Batch Profiler task. This task runs image processing that includes page identification, image cleaning, image rotation, find fingerprints, and full character recognition.
- [Verifying invoice data in Datacap Desktop](#)
 You can verify the invoice data by running the APT Verify task on Datacap Desktop to make sure that the sample invoice data was accurately captured and recognized by Datacap Accounts Payable. The invoice image is displayed next to the invoice data fields to help you identify and correct data problems before you export the batches.
- [Exporting invoice images on Datacap Desktop](#)
 The APT Export task runs on Datacap Desktop to finish invoice image processing, save the invoice information, and export the field data into XML files. You can store these files in your Microsoft Windows environment.

Parent topic: [Accounts Payable application](#)

Scanning invoice images on Datacap Desktop

You can run the APT VScan task to create a batch of invoices from the sample images that are provided with Datacap Accounts Payable. You can also set up an APT VScan task to scan images from a scanner or MFP device.

About this task

The APT VScan task uses Datacap Desktop to select a job and run a virtual scan to bring the invoice images into Datacap Accounts Payable. Datacap Accounts Payable picks up the images from the \APT\images\Input folder and creates a Datacap batch of document images for processing. Datacap Accounts Payable assigns a batch ID and creates a folder on Datacap where the documents are stored.

Procedure

Follow this procedure to scan invoice images on Datacap Desktop.

1. On the developer workstation from the Start menu, select Datacap Clients > Datacap Desktop.

2. Log in using the default User, Password, and Station (`admin`, `admin`, and `1`).
 3. Select APT from the Applications menu.
 4. Select APT Demo VScan from the Task Shortcuts menu.
 5. At the APT Demo VScan batches - Datacap Desktop window, click Run Pending.
 6. Select the job that you want to run.
 - o Demo: Uses the included single and multipage invoices
 - o Demo-Dot Matrix: Uses your dot-matrix invoices that you put in the `/input` directory in place of the included images
 - o Demo-Flex ID: Enables a task after the scanning to manually identify the invoices, typically used when a batch of multi-page invoices were scanned without document separator sheets
- Datacap Desktop scans the images for the selected batch type and displays `Batch<batch ID>finished with a status of finished`. The new batch is displayed in the Batch View frame.
7. Click OK to close the APT Demo VScan batches - Datacap Desktop window.

Datacap Accounts Payable retrieved the images from the `Datacap\APT\images\Input` folder and created a Datacap batch of the document images to be processed. Datacap assigned a batch ID and created a uniquely numbered folder with the name of the batch ID in the `Datacap\APT\batches` folder. This folder contains all of the document image-processing results for this batch of invoices, including the newly created `rrsvscan.xml` file.

8. Click Run Pending to scan the images for the next batch of invoices.
9. When you complete all of your batches, close the APT Demo VScan - Datacap Desktop window.

Parent topic: [Datacap Accounts Payable client procedures](#)

Preparing invoice images for processing on Datacap Desktop

You run the APT FlexID task on Datacap Desktop to prepare the invoice images for the processing that takes place in the Batch Profiler task. You can change the image size, orientation, sequence in the batch, and page type settings.

Procedure

Follow this procedure to prepare invoice images for processing on Datacap Desktop.

1. On the developer workstation from the Start menu, select Datacap Clients > Datacap Desktop. The Datacap Desktop login window opens.
2. Log in using the default User, Password, and Station (`admin`, `admin`, and `1`).
3. Select the APT application from the Applications menu.
4. Select the FlexID task from the Task Shortcuts menu. The images are displayed in frames on the window.
5. Click the drop-down menu for the image and designate the page type of the image.
6. Click Next to move to the next image, if all of the images are not displayed on the window.
7. Click Go to End to move to the last image in the batch.
8. Change the sequence of images in the batch by clicking the thumbnail of the image and moving it to the wanted page.
9. Click Add Before to add a separator page before the selected image.
10. Click Add After to add a separator page after the selected image.
11. Click Hold to put the processing of the batch on hold.
12. Complete the settings for the images in the batch and click Save and Exit, the APT Demo FlexID window closes. A `FlexID.xml` file that contains the Page Types you assigned in the batch ID numbered folder in `\APT\batches`.
13. When you complete all of your batches, close the APT Demo FlexID - Datacap Desktop window.

Processing invoice images on Datacap Desktop

After you scan the invoice images and prepare them for processing, use Datacap Desktop to run the APT Batch Profiler task. This task runs image processing that includes page identification, image cleaning, image rotation, find fingerprints, and full character recognition.

About this task

The APT Batch Profiler task uses Rulerunner to process the images. When the invoice images are processed, a fingerprint of the image is created. Fingerprints are used by Datacap to identify the vendor that is associated with the invoice. Fingerprints include the location of the data fields on the image to be captured.

Procedure

To process invoice images on Datacap Desktop:

1. On the developer workstation from the Start menu, select Datacap Clients > Datacap Desktop.
2. Log in using the default User, Password, and Station (`admin`, `admin`, and `1`).
3. Select APT from the Applications menu.
4. Select Batch Profiler from the Task Shortcuts menu. The APT Demo Batch Profiler window opens with a progress bar. Rulerunner is running a number of processing-intensive tasks, including PageID, ImageFix, CreateDocs, Recognize, FindFingerprint, Locate, and lookups. When processing is done, it displays `Batch <batch ID> finished with a status of finished.`
5. Click OK. The images were organized into documents, copies were made of the original images, fingerprint (CCO). The Batch Profiler.xml page file, data files, and .txt files of the raw recognition results are created in the uniquely numbered folder for the batch in `Datacap\APT\batches`.

Verifying invoice data in Datacap Desktop

You can verify the invoice data by running the APT Verify task on Datacap Desktop to make sure that the sample invoice data was accurately captured and recognized by Datacap Accounts Payable. The invoice image is displayed next to the invoice data fields to help you identify and correct data problems before you export the batches.

Procedure

Follow this procedure to verify invoice data in Datacap Desktop. For information on verifying pages, see [Verifying pages by using Datacap Desktop](#).

1. On the developer workstation from the Start menu, select Datacap Clients > Datacap Desktop.
2. Log in using the default User, Password, and Station (`admin`, `admin`, and `1`).
3. Select the APT application from the Applications menu.
4. Select Verify from the Task Shortcuts menu. The APT Verify Batches - Datacap Desktop window opens with a list of batches to verify.

All of the invoices are set to display in the Verify window. In a production environment, you can display every invoice to the Verify operator or you can display only the invoices where recognition results are low confidence and data validation rules are violated.

5. Double-click the batch that you want to verify.

The Verify window displays information for the first invoice in the batch and the details for the first line item on that invoice. It displays fields, recognized data, snippets, and a number of buttons, with invoice header information displayed on top, and line item detail on the bottom.

The Image View panel displays the scanned invoice image.

The Batch View panel displays a list of the images that are included in the batch.

When there are errors or low confidence results, the cursor is placed in the first problem field and the field is outlined in red. In addition, the background color of a data field indicates:

- o Blue: No data validation errors or low confidence recognition results
- o Yellow: Low confidence recognition results, low confidence characters are displayed in red
- o Red: Data validation error

When a field fails validation with red background, you must correct the data before you can complete the batch. When the recognition results for a field are low confidence with yellow background, you can update the contents of the field. You can also move out of the field to change the background color to blue, or you can ignore the low confidence condition and press Next Problem to go to the next document in the batch.

6. The following table describes the features of the APT Verify window.

On the	To	Do this action
Verify window	Move to the next data field	Press Tab
Verify window	Fill empty field with data by using Click N Key	Click in an empty data entry field, then do one of the following actions: <ul style="list-style-type: none"> o Point at the field value on the document image and click it. o Click and drag the mouse to draw a box around the existing field data on the image. o Press and hold Alt, click, and drag the mouse to draw a box that encompasses more blank space to leave room for extra characters. o Press and hold Shift, click a field that was already defined, and drag the mouse to extend the field.
Verify window	Display the detail for the selected invoice field	Click Find Details
Verify window	Calculate missing numeric values by using existing values and automatically fill blank numeric fields	Click Calculate Blank
Verify window	Look up and select vendor (either when blank or when incorrect vendor is displayed)	Click the Vendor field title to display the Looking up data for vendor dialog. Search for and enter the wanted vendor data

On the	To	Do this action
Verify window	Open the Purchase Order Line Reconciliation window to match or review matched line items	Click POLR
Verify window	Identify type of invoice	Select from the Invoice Type dropdown menu list
Verify window	Create new fingerprint for this invoice	Click Add_New_Fingerprint and select Yes
Verify window	Add a line before current, displayed line on the invoice	Click Insert
Verify window	Add a line after current, displayed line on the invoice	Click Append
Verify window	Delete the current, displayed line from the invoice	Click Delete
Process tab	Move to next low confidence character	Click Next LC
Process tab	Complete the batch after verification	Click Submit
Process tab	Move to the previous problem document in the batch	Click Previous Page
Process tab	Move to the next problem document in the batch	Click Next Problem
Process tab	Display the previous document in the batch.	Click Prev Image
Process tab	Display the next document in the batch	Click Next Image
Process tab	Toggle the Image View frame	Click Document Images
Process tab	Toggle the Batch View frame	Click Batch Structure
Process tab	Display or hide the scanned image	Click Hide Toolbar
Process tab	Invoke validation rules for entire invoice	Click Run Validations
Snippets tab	Enlarge or reduce the size of the data in the fields on the Verify window	Click Zoom In or Zoom Out
Snippets tab	Move the data in the fields vertically	Click Move Up or Move Down
Snippets tab	Move the data in the fields horizontally	Click Move Left or Move Right
Snippets tab	Restore the data in the fields to the original position	Click Restore Snippets
Image Display tab	Change the display of the image in the Image View to quarter view	Click Quarter View
Image Display tab	Fit the display of the image to the width or the height of the Image View frame	Click Fit to Width or Fit to Height
Image Display tab	Display the image in its original size in the Image View frame	Click Whole Image
Image Display tab	Highlight the captured fields on the image in the Image View frame	Click Captured Fields

On the	To	Do this action
Image Display tab	Highlight the words or lines from the Fingerprint CCO file in the Image View frame	Click CCO Words or CCO Lines
Image Display tab	Rotate the image in the Image View frame	Click Rotate 90 or Rotate 270
Exceptions tab	Mark the document for review	Click Review
Exceptions tab	Mark the document for rescan	Click Rescan
Exceptions tab	Mark the document for deletion	Click Delete
Exceptions tab	Reset the document to its original state	Click Reset

7. Click POLR to run Purchase Order Line Reconciliation to match invoice line items with corresponding purchase order line items. The Purchase Order Line Reconciliation window displays the matched line items as well as those line items that did not match.
8. Resolve any line items that did not match in POLR.
9. Review and click Run Validations on each sample document. When the data is validated, a *Task Profile succeeded* message is displayed.
10. Click OK to go to the next document in the batch. When you finish processing the last invoice in the batch, a message box indicates that no additional problems were found.
11. Click Submit to finish the batch. A message box is displayed indicating the task is done.
12. Complete the remaining batches.

- [APT Verify window instructions](#)

You can process unknown invoices and resolve other issues that might arise.

- [Permission denied: LoadPicture Error](#)

This situation occurs when inadequate permissions are granted to users and to a graphic image that is stored in the APT dco folder.

- [Changes to Pre-Existing Zone Positions Are Ignored](#)

This situation occurs when an invoice on which, when processed the first time, the field zones were not set up properly. A subsequent attempt to change the existing zones is ignored.

- [Identify Fingerprint ID Associated with Problem Invoice](#)

When fields on an invoice are not zoned correctly, the data from the invoice is not captured when the invoice is displayed in the Verify window.

Parent topic: [Datacap Accounts Payable client procedures](#)

Related information:

[Keyboard shortcuts for the Datacap Desktop verify task](#)

APT Verify window instructions

You can process unknown invoices and resolve other issues that might arise.

About this task

These instructions use the following terms:

Matched/Unmatched: Datacap Accounts Payable processing finds a fingerprint that it thinks matches an invoice; this match might be correct or incorrect.

Known/Unknown: Datacap Accounts Payable has or does not have the vendor in a lookup database; Datacap Accounts Payable has or does not have a fingerprint for the invoice.

- [Handling an unknown invoice when the vendor is unknown](#)
This situation occurs when you handle an unknown invoice for an unknown vendor. For example, when your company is invoiced by a new vendor before that vendor is added to the lookup database.
- [Identifying detail lines on an unknown invoice](#)
This situation occurs when you want to identify where the first line item is located. Or where each of the fields in the line is located or what type of information they contain on unknown invoices.
- [Processing multiple unknown invoices for a known vendor in a single batch](#)
This situation occurs when you process multiple unknown invoices in a single batch for a known vendor. The vendor changed the layout of their invoices. Or you are processing previously unprocessed invoices for a vendor who was recently added to the lookup database.
- [Associating a vendor with a fingerprint](#)
This situation occurs when you correct the association between a vendor and a fingerprint. Then, you correct the data when Datacap Accounts Payable matches an invoice with the wrong fingerprint. This scenario can happen when two vendors use nearly identical invoice formats.
- [Capturing fields on a rotated image](#)
This situation occurs when you process an invoice that you rotated.

Parent topic: [Verifying invoice data in Datacap Desktop](#)

Handling an unknown invoice when the vendor is unknown

This situation occurs when you handle an unknown invoice for an unknown vendor. For example, when your company is invoiced by a new vendor before that vendor is added to the lookup database.

About this task

When Datacap Accounts Payable processes the invoice, the Vendor Name is blank, and when you click Lookup Vendor, the vendor is not listed.

If Microsoft Outlook is installed on the Workstation, you can route the invoice to a knowledge worker for handling. The APT Settings.ini file must be updated with real email addresses.

Procedure

To handle an unknown invoice when the vendor is unknown:

1. Mark the invoice for supervisor review:
 - a. Click Exceptions > Review.
 - b. Enter the relevant comments and click OK.
 - c. During the Export task, an email that contains this invoice image is sent to the email address specified by the Review= parameter in the Settings.ini file. The file is not exported
2. Delete the invoice from the batch:
 - a. Click Exceptions > Delete.
 - b. Click Yes at the message box to confirm the deletion of the document.
 - c. During the Export task, an email that contains this invoice image is sent to the email address specified by the Delete= parameter in the Settings.ini file.
3. If Microsoft Outlook is not installed on the Workstation, you can put the entire batch on hold and alert a Supervisor to the problem
 - a. Select File > Quit Task.
 - b. Click OK to put batch on hold.
 - c. Alert the Supervisor to the problem.

Parent topic: [APT Verify window instructions](#)

Identifying detail lines on an unknown invoice

This situation occurs when you want to identify where the first line item is located. Or where each of the fields in the line is located or what type of information they contain on unknown invoices.

About this task

When an unknown invoice is processed, the Lineitem counter displays 0(0) - line 0 of 0.

Procedure

To identify detail lines on unknown invoices:

1. Click Insert to add an empty line.
2. Use Click N Key to identify where each of the fields is in the first line, then click Find Details to locate the remaining lines. The Lineitem counter displays an actual line count. The details of the data that is captured for all of the lines is displayed in a scrollable area on the main screen.
3. Close the window, and press Alt+V to validate, and ensure that the invoice data passes all validations.

Parent topic: [APT Verify window instructions](#)

Processing multiple unknown invoices for a known vendor in a single batch

This situation occurs when you process multiple unknown invoices in a single batch for a known vendor. The vendor changed the layout of their invoices. Or you are processing previously unprocessed invoices for a vendor who was recently added to the lookup database.

About this task

When Datacap Accounts Payable processes the first unknown invoice, the Vendor Name is blank.

Procedure

To process the invoice as an unknown invoice:

1. Enter the first few letters of the vendor name and the vendor postal code and click Lookup Vendor.
2. Populate the data fields by using Click N Key, Find Details and press Alt+V. The invoice data is validated, ensure that the data passes all of the validations. Use Click N Key to populate the fields so that the data locations on the images can be stored as part of the fingerprint.
3. When the Lineitem counter displays an actual count, click View Details to review the data that is captured for all of the lines in a separate window.
4. Close the window and press Alt+V to validate. Ensure that the invoice data passes all validations. When the next invoice in that batch for the same vendor is displayed in the Verify window. APT matches it to the earlier invoice that passed validation. APT displays Sticky Fingerprint Available next to Vendor Lookup.
5. Click Sticky Fingerprint Available to locate and populate the fields automatically.

Parent topic: [APT Verify window instructions](#)

Associating a vendor with a fingerprint

This situation occurs when you correct the association between a vendor and a fingerprint. Then, you correct the data when Datacap Accounts Payable matches an invoice with the wrong fingerprint. This scenario can happen when two vendors use nearly identical invoice formats.

About this task

When Datacap Accounts Payable processes the invoice, it displays the wrong vendor and the data fields are populated with incorrect data.

Procedure

To associate the vendor with the fingerprint:

1. Enter the first few letters of the vendor name and the vendor postal code, click Lookup Vendor and select the correct vendor.
2. Ensure that all of the data fields are populated correctly by using Click N Key, Find Details and press Alt+V to validate.
3. Click New Fingerprint. Use Click N Key to populate the fields so the data locations on the images can be stored as part of the fingerprint.

Parent topic: [APT Verify window instructions](#)

Capturing fields on a rotated image

This situation occurs when you process an invoice that you rotated.

About this task

The Click N Key feature no longer works.

Procedure

To capture fields on a rotated image:

1. Manually key the data into the fields from the image and press Alt+V to validate.
2. Even though the invoice passes all validations, a fingerprint is not automatically created by Datacap Accounts Payable. The next invoice from that vendor is treated as an unknown invoice.

Parent topic: [APT Verify window instructions](#)

Permission denied: LoadPicture Error

This situation occurs when inadequate permissions are granted to users and to a graphic image that is stored in the APT dco folder.

About this task

The Permission denied: LoadPicture Error is returned when you run the Verify task in APT or Flex.

This issue depends on your user permissions and if you are logged on to Windows as a Restricted User. You might get this message when you run the Verify task in APT or Flex.

Procedure

To grant user permission:

1. Open Windows Explorer and open the application dco_ folder. This folder is usually on Datacap Server.
2. Right-click the fprintnew.bmp file and select Properties.
3. Click the Security tab and add the user.
4. Grant the user Full Control to the fprintnew.bmp file.

Parent topic: [Verifying invoice data in Datacap Desktop](#)

Changes to Pre-Existing Zone Positions Are Ignored

This situation occurs when an invoice on which, when processed the first time, the field zones were not set up properly. A subsequent attempt to change the existing zones is ignored.

About this task

When Datacap Accounts Payable processes an invoice from a known vendor, data is missing or captures the wrong data.

When an invoice is processed for the first time, an incorrect fingerprint for the invoice can be created. While Datacap Accounts Payable supports adding missing fields to an existing fingerprint, it does not support changing existing field positions.

Procedure

To correct a fingerprint with incorrectly zoned fields, do the following tasks:

1. [Identify Fingerprint ID Associated with Problem Invoice](#)
2. [Delete fingerprints](#)

Parent topic: [Verifying invoice data in Datacap Desktop](#)

Identify Fingerprint ID Associated with Problem Invoice

When fields on an invoice are not zoned correctly, the data from the invoice is not captured when the invoice is displayed in the Verify window.

About this task

The fields on an invoice are not zoned correctly.

Procedure

To identify the Fingerprint ID associated with the problem invoice:

1. With the invoice is displayed in the Verify window, make a note of the name of the vendor that is displayed on the invoice. Note the Batch ID and document number that is found in the lower right portion of the status bar.
2. Using Windows Explorer, go to the application Batches folder, and open the folder that contains the batch.
3. Using a text editor such as Notepad, open the Verify.xml file.

4. Locate the portion of the file that is associated with the Batch ID and document number.
5. Locate the TemplateID tag and the Fingerprint ID.
6. Use the Fingerprint ID to identify and delete the fingerprint by using Datacap Studio or the Fingerprint Maintenance Tool.

Parent topic: [Verifying invoice data in Datacap Desktop](#)

Exporting invoice images on Datacap Desktop

The APT Export task runs on Datacap Desktop to finish invoice image processing, save the invoice information, and export the field data into XML files. You can store these files in your Microsoft Windows environment.

Procedure

To export invoice images on Datacap Desktop:

1. On the developer workstation from the Start menu, select Datacap Clients > Datacap Desktop.
2. Log in using the default User, Password, and Station (`admin`, `admin`, and `1`).
3. Select APT from the Applications menu.
4. Select Export from the Task Shortcuts menu.
 - Export outputs the field data to XML files. The XML files and the image files are now ready to be input into a back-end application.
 - An Export XML file was added to the uniquely numbered folder of the batch in `Datacap\APT\export`. The PDF and XML files in the export folder correspond to each Document ID in the batch.
 - Export saves information about unknown invoices that were successfully validated and creates new fingerprints. The next time APT detects the same invoice. APT can identify the vendor and extract the field data by using a process that is called Intellocate.

During Verify, you can mark an invoice for Review or Rescan, or delete an invoice. If you did not configure email notifications, the PDF versions of the marked or deleted images stay in the batch numbered folder in `Datacap\APT\batches`. The images and information about the images are not exported. You can configure email notifications for invoice images that are marked or deleted. In this case, Export sends an email with the marked or deleted image that is attached as a PDF file.

5. Use Windows Explorer to go to and open the resulting PDFs for each invoice. You can open the `Export.xml` file to view the data that is captured from the invoices.

Parent topic: [Datacap Accounts Payable client procedures](#)

Datacap Accounts Payable Web Client procedures

You can use the Datacap Accounts Payable application to process pre-scanned images by running the default shortcuts on the Datacap Web Client.

You must start the Datacap Server Service and the Datacap Web Client before you begin.

Table 1. Sequence in which to run the default shortcuts

This shortcut	Description
iVScan	Process sample, pre-scanned images
Upload	Move scanned images to the Datacap Server Service for further processing

This shortcut	Description
Verify	Perform data validation

Important: To complete processing, the APT Batch Profiler and Export tasks must be run manually from the Datacap Desktop client. In a client/server environment, the APT Batch Profiler and Export tasks can be run automatically as background tasks by either Datacap Desktop or Rulerunner.

- [Review default installation results](#)
You can review the Datacap Client groups, users, and stations that are installed by the default installation process. This review helps you become familiar with the panels that are used to add new groups, users, and stations and assign application-specific permissions.
- [Logging in to Datacap Accounts Payable on Datacap Web Client](#)
You need the web address and the IP address of the Datacap Web Client server to log in to Datacap Accounts Payable on the Datacap Web Client.
- [Scanning invoice images on Datacap Web Client](#)
You scan sample invoice images on Datacap Web Client by running the APT iVScan task.
- [Uploading invoice images on Datacap Web Client](#)
You upload the invoice images to the batch server on Datacap Web Client by running the APT Upload task.
- [Processing invoice images on Datacap Desktop](#)
After you scan the invoice images and prepare them for processing, use Datacap Desktop to run the APT Batch Profiler task. This task runs image processing that includes page identification, image cleaning, image rotation, find fingerprints, and full character recognition.
- [Verifying invoice data on Datacap Web Client](#)
You can verify the invoice data on Datacap Web Client to ensure that the invoice data was accurately captured and recognized by Datacap Accounts Payable. The APT Verify task displays the invoice image next to the invoice data fields. So you can identify and correct potential data problems before you export the batches.
- [Exporting invoice images on Datacap Desktop](#)
The APT Export task runs on Datacap Desktop to finish invoice image processing, save the invoice information, and export the field data into XML files. You can store these files in your Microsoft Windows environment.

Parent topic: [Accounts Payable application](#)

Review default installation results

You can review the Datacap Client groups, users, and stations that are installed by the default installation process. This review helps you become familiar with the panels that are used to add new groups, users, and stations and assign application-specific permissions.

Prerequisites

1. Ensure that you meet the installation and configuration prerequisites for the Datacap.
 2. Complete the instructions that are provided to install all of the components on a single computer on which you have Administrative rights.
 3. Complete the instructions to install and configure all of the components on the Server and the Workstation. And to copy the application to the Server, and to complete the Server configuration.
 4. Obtain the password from Datacap that you must use when you log in as the Datacap admin user.
- [Starting Datacap Accounts Payable on Datacap Web Client](#)
You can run the Datacap Accounts Payable application on Datacap Web Client. If you logged in to Datacap Web Client in the past, the information you previously entered is displayed.

- [Viewing the default user security group on Datacap Web Client](#)
The default Datacap installation and configuration sets up some Datacap groups for the Datacap Accounts Payable application. You can view the default user security group that is delivered as part of Datacap Web Client APT.
- [Viewing the default user permissions on Datacap Web Client](#)
The default Datacap installation and configuration sets up some Datacap user permissions for the Datacap Accounts Payable application. You can view the default user permissions that are delivered as part of Datacap Web Client APT.
- [Viewing the default station permissions on Datacap Web Client](#)
The default Datacap installation and configuration sets up some Datacap station permissions for the Datacap Accounts Payable application. You can view the default station permissions that are delivered as part of Datacap Web Client APT.

Parent topic: [Datacap Accounts Payable Web Client procedures](#)

Related concepts:

[Installation and configuration prerequisites](#)

Starting Datacap Accounts Payable on Datacap Web Client

You can run the Datacap Accounts Payable application on Datacap Web Client. If you logged in to Datacap Web Client in the past, the information you previously entered is displayed.

Procedure

To start Datacap Accounts Payable on Datacap Web Client:

1. Start Internet Explorer and open the Datacap Web Client home page:
 - If the server is running on the same computer, use `http://localhost/tmweb.net`
 - If the server is running on a different computer, use `http://tmweb_server/tmweb.net`
2. Log in to the APT application.
3. Enter `admin` in the User ID and Password fields.
4. In the Station field:
 - Enter `1` if the server is running on the same computer.
 - Enter `remote` if the server is running on a different computer.
5. Click OK
6. The APT main window opens with the Operations pane open.

Parent topic: [Review default installation results](#)

Viewing the default user security group on Datacap Web Client

The default Datacap installation and configuration sets up some Datacap groups for the Datacap Accounts Payable application. You can view the default user security group that is delivered as part of Datacap Web Client APT.

Procedure

To view the default user security group:

1. Start Internet Explorer and open the Datacap home page:
 - If the server is running on the same computer, use `http://localhost/tmweb.net`
 - If the server is running on a different computer, use `http://tmweb_server/tmweb.net`
2. Log in to the APT application.

3. Click the Administrator tab and select Groups.
4. Select Scanners from the list.
5. Expand the Privileges, Permissions, and Users branches to view the options that are set for this group. The Scanners group is granted permission to do the iVScan and Upload tasks for the web-based jobs. These jobs are the Web Demo, Web Demo Dot-Matrix, Web Main, and Web Main Dot-Matrix jobs.
6. If you are finished viewing groups, close the Datacap Client window.

Parent topic: [Review default installation results](#)

Viewing the default user permissions on Datacap Web Client

The default Datacap installation and configuration sets up some Datacap user permissions for the Datacap Accounts Payable application. You can view the default user permissions that are delivered as part of Datacap Web Client APT.

Procedure

To view the default user permissions:

1. Start Internet Explorer and open the Datacap home page:
 - o If the server is running on the same computer, use `http://localhost/tmweb.net`
 - o If the server is running on a different computer, use `http://tmweb_server/tmweb.net`
2. Log in to the APT application.
3. Click the Administrator tab and select Users.
4. Select the scan1 user from the list.
5. Expand the Privileges, Permissions, and Users branches to view the options that are set for this user. The scan1 user is part of the Scanners group, and is granted the permissions that were granted to the group.
6. If you are finished viewing users, close the Datacap Client window.

Parent topic: [Review default installation results](#)

Viewing the default station permissions on Datacap Web Client

The default Datacap installation and configuration sets up some Datacap station permissions for the Datacap Accounts Payable application. You can view the default station permissions that are delivered as part of Datacap Web Client APT.

Procedure

To view the default station permissions:

1. Start Internet Explorer and open the Datacap home page:
 - o If the server is running on the same computer, use `http://localhost/tmweb.net`
 - o If the server is running on a different computer, use `http://tmweb_server/tmweb.net`
2. Log in to the APT application.
3. Click the Administrator tab and select Users.
4. Select the remote station from the list.
5. Expand the Privileges, Permissions, and Users branches to view the options that are set for this station. The remote station is permissions to run iVScan, Upload, and Verify tasks.
6. If you are finished viewing stations, close the Datacap Client window.

Parent topic: [Review default installation results](#)

Logging in to Datacap Accounts Payable on Datacap Web Client

You need the web address and the IP address of the Datacap Web Client server to log in to Datacap Accounts Payable on the Datacap Web Client.

Before you begin

Before logging in to Datacap Accounts Payable, [start or ensure the Datacap Server Service is started](#).

Procedure

To log in to Datacap Accounts Payable on the Datacap Web Client:

1. Open Internet Explorer, and enter the address of the Datacap Web Client server followed by the alias for the website <http://localhost/tmweb.net>. After a pause, the Datacap Web Client Login window opens.
Tip: Turn off the Internet Explorer pop-up blocker the first time that the Login page opens.
2. Open Internet Explorer, and enter the IP address of the Datacap Web Client server followed by the alias for the website. For example, <http://127.0.0.1/tmweb.net>, <http://WebServerName/tmweb.net>, or <https://WebServerName/tmweb.net>. After a pause, the Datacap Web Client Login window opens.
Tip: Turn off the Internet Explorer pop-up blocker the first time that the Login page opens.
3. On the Datacap Web Client Login page, enter the following values.

Table 1. Datacap Web Client login entries

Field	Value
Application	APT
User ID	admin
Password	admin
Station	remote

4. Click Login. The Datacap Accounts Payable main window opens with the Operations pane and Available shortcuts.

Parent topic: [Datacap Accounts Payable Web Client procedures](#)

Scanning invoice images on Datacap Web Client

You scan sample invoice images on Datacap Web Client by running the APT iVScan task.

Procedure

To scan images on Datacap Web Client:

1. In Internet Explorer, on the Operations tab, click the iVScan shortcut. The list of available jobs is displayed.
2. Click the Web Demo shortcut. The Web Demo Scan page opens with the batch number assigned to this batch of documents in the header. By default, the Multiple files option is selected.
3. The Scanned Image Folder field on the page indicates where the pre-scanned images are temporarily placed. If you do not have permission to write to this location, change this path to one to which you can write.

4. In the Source Directory field, click Browse and go to the folder where the pre-scanned images are installed. The default location is in the application folder \Datacap\APT\Images\Input.
5. Select the first file in the folder and click Open. The Web Demo Scan page opens with the path to the Source Directory folder.
6. You do not have to process all of the sample invoices. You can change the number in the Expected pages field to the number of images you want to process. Then, click Scan. The pages are scanned and images of the pages are displayed. The batch-specific name of the folder is appended to the Scanned Image folder path and an End of scan message is displayed.
7. Click OK.
8. The following table describes some of the things you can do on the Web Demo Scan dialog:

Table 1. Functions that you can do on the Web Demo Scan dialog

To	Do this step
Change sequence of pages	Select either the image or the name of the image that you want to move. Then, click either the MoveUp or MoveDown arrow
Delete all pages	Click Remove All
Delete a single page	Select the image that you want to delete, then click Remove
Enlarge scanned images	Click Zoom in
Shrink scanned images	Click Zoom out
Rotate an image	Click Rotate 90
Insert the image before the first image in the batch	Click Insert before
Complete the Scan Task processing of the batch	Click Done
Cancel the Scan Task processing of the batch; deletes batch and all batch data from system	Click Cancel
Put the batch on hold and end processing of the batch; batch remains available to same user	Click Hold

9. After all of the images are scanned, click Done. A message box is displayed indicating the Batch finished with a status of finished.
10. Click OK. A message box asks if you want to continue processing.
11. Click Stop. The message box closes and the Operations pane opens.
12. When the iVScan task finishes successfully, the source images are collected into a batch. These images are copied to the uniquely numbered folder of the batch in the C:\Datacap\Scan folder.
13. Click the Monitor tab to view information about the batch.
14. Press F5 to refresh the information. The Job, Task, and Status for this batch are updated and have a status of Web Demo, Upload, pending.

Parent topic: [Datacap Accounts Payable Web Client procedures](#)

Uploading invoice images on Datacap Web Client

You upload the invoice images to the batch server on Datacap Web Client by running the APT Upload task.

Procedure

To upload invoice images on Datacap Web Client:

1. In Internet Explorer, on the Operations tab, click the Upload shortcut. The Upload window opens. The transferring the batch to server message is displayed followed by another message that indicates the batch finished with a status of finished.
2. Click OK. A message box asks if you want to continue processing.
3. Click Stop. The Operations pane opens.
4. When the Upload task finishes successfully, the scanned images are transferred to the uniquely numbered folder for the batch in \Datacap\APT\batches. The VScan.xml and Upload.xml files were created.
5. Click the Monitor tab to view information about the batch.
6. Press F5 to refresh the information. The Job, Task, and Status for this batch were updated and are now Web Demo, Batch Profiler, pending.
7. Run the next task, the APT Batch Profiler task, manually from the Datacap Client APT, such as Datacap Desktop.

Parent topic: [Datacap Accounts Payable Web Client procedures](#)

Processing invoice images on Datacap Desktop

After you scan the invoice images and prepare them for processing, use Datacap Desktop to run the APT Batch Profiler task. This task runs image processing that includes page identification, image cleaning, image rotation, find fingerprints, and full character recognition.

About this task

The APT Batch Profiler task uses Rulerunner to process the images. When the invoice images are processed, a fingerprint of the image is created. Fingerprints are used by Datacap to identify the vendor that is associated with the invoice. Fingerprints include the location of the data fields on the image to be captured.

Procedure

To process invoice images on Datacap Desktop:

1. On the developer workstation from the Start menu, select Datacap Clients > Datacap Desktop.
2. Log in using the default User, Password, and Station (admin, admin, and 1).
3. Select APT from the Applications menu.
4. Select Batch Profiler from the Task Shortcuts menu. The APT Demo Batch Profiler window opens with a progress bar. Rulerunner is running a number of processing-intensive tasks, including PageID, ImageFix, CreateDocs, Recognize, FindFingerprint, Locate, and lookups. When processing is done, it displays `Batch <batch ID> finished with a status of finished.`
5. Click OK. The images were organized into documents, copies were made of the original images, fingerprint (CCO). The Batch Profiler.xml page file, data files, and .txt files of the raw recognition results are created in the uniquely numbered folder for the batch in Datacap\APT\batches.

Parent topic: [Datacap Accounts Payable Web Client procedures](#)

Verifying invoice data on Datacap Web Client

You can verify the invoice data on Datacap Web Client to ensure that the invoice data was accurately captured and recognized by Datacap Accounts Payable. The APT Verify task displays the invoice image next to the invoice data fields. So you can identify and correct potential data problems before you export the batches.

Procedure

To verify invoice data on Datacap Web Client:

1. In Internet Explorer, on the Operations tab, click the Verify shortcut. The Verification panel opens with the first image displayed.
2. The following table describes some of the things you can do on the Verification panel:

Table 1. Functions that you can do on the Verification panel

To	Do this step
Move to the next data field	Press Tab
Move to the next low confidence character	Click Next LC
Look up and select vendor (either when blank or when incorrect vendor is displayed)	Enter at least the first character of the vendor name and the Remittance_Zip. Then, click the Vendor link over the Vendor Name snippet to open the lookup window
Fill empty field with data by using Click N Key	Click in the empty data entry field, then do one of the following steps: Point at the field value on the image and click it, or, Click and drag the mouse to draw a box around the existing field data on the image
Complete processing of image and display next image	Click Submit
Mark the document for deletion, rescanning, or review	Select appropriate choice from the Routing Instructions list
Display the first page of the next document in the batch without submitting or revalidating data	Click Next Page
Display the first page of the previous document in the batch without submitting or revalidating data	Click Previous Page
Display the first page, next page, previous page, or last page of the current document	Click < > < or > next to the Page_No label
Display the details of the first line item, next line item, previous line item, or last line item	Click < > < or > next to the Lineitem label
For all detail lines, locate all field values for all fields	Click Find Details
Calculate missing numeric values by using existing values and automatically fill blank numeric fields	Click Calculate Blank

To	Do this step
Move to the next problem document in the batch	Click Next Problem
Move to the previous problem document in the batch	Click Previous Problem
Identify the type of invoice	Select from Invoice Type dropdown list
Display snippet that is associated with a field	Select Disp snip
Create new fingerprint for this invoice	Select Yes from Add New Fingerprints dropdown list
Put entire batch on hold	Click Hold
Override an error	Select Override
Zoom in or out, resize the image	Right-click on the image and select the appropriate action

3. When you finish the last document, click Submit. A message box is displayed indicating that all of the documents are complete and you are prompted to finish the batch.
4. Click OK. A message box asks if you want to continue processing.
5. Click Stop. The Operations pane displays.
6. When the Verify task finishes successfully, page XML file were updated. For example, when low confidence characters were changed during Verify. A Verify.xml file was created in the uniquely numbered folder of the batch in \Datacap\APT\batches.
7. Click the Monitor tab to view information about the batch.
8. Press F5 to refresh the information. The Job, Task, and Status for this batch were updated and are now Web Demo, Batch Profiler, pending.
9. Run the next task, the APT Export task, manually from the Datacap Client APT, such as Datacap Desktop.

Parent topic: [Datacap Accounts Payable Web Client procedures](#)

Exporting invoice images on Datacap Desktop

The APT Export task runs on Datacap Desktop to finish invoice image processing, save the invoice information, and export the field data into XML files. You can store these files in your Microsoft Windows environment.

Procedure

To export invoice images on Datacap Desktop:

1. On the developer workstation from the Start menu, select Datacap Clients > Datacap Desktop.
2. Log in using the default User, Password, and Station (admin, admin, and 1).
3. Select APT from the Applications menu.
4. Select Export from the Task Shortcuts menu.
 - o Export outputs the field data to XML files. The XML files and the image files are now ready to be input into a back-end application.
 - o An Export XML file was added to the uniquely numbered folder of the batch in Datacap\APT\export. The PDF and XML files in the export folder correspond to each Document ID in the batch.
 - o Export saves information about unknown invoices that were successfully validated and creates new fingerprints. The next time APT detects the same invoice. APT can identify the vendor and extract the field data by using a process that is called Intellocate.

During Verify, you can mark an invoice for Review or Rescan, or delete an invoice. If you did not configure email notifications, the PDF versions of the marked or deleted images stay in the batch numbered folder in Datacap\APT\batches. The images and information about the images are not exported. You can configure email notifications for invoice images that are marked or deleted. In this case, Export sends an email with the marked or deleted image that is attached as a PDF file.

5. Use Windows Explorer to go to and open the resulting PDFs for each invoice. You can open the Export.xml file to view the data that is captured from the invoices.

Parent topic: [Datacap Accounts Payable Web Client procedures](#)

Medical Claims application

Medical Claims captures and validates data from CMS-1500 professional claims and UB-04 institutional claims in the United States.

About this task

By default, the Medical Claims application captures and exports the complete set of data from claim forms. However, you customize the application to remove undesired fields, validations, and data that gets exported as needed for your specific claims processor.

Medical Claims jobs include:

INST Red

For use with red institutional medical claim forms.

INST Black

For use with black institutional medical claim forms.

PROF Red

For use with red professional medical claim forms.

PROF Black

For use with black professional medical claim forms.

Remember: You must install and configure Datacap Web (Datacap Web Server and Datacap Web Client) in order to use the Medical Claims web application.

Attention: The Medical Claims application on the Datacap Web Client only includes Web Scan jobs with the ability to import previously scanned images from disk. However, you can configure the application for physical scanning by manually adding tasks to scan paper claims from the Datacap Web Client.

Restriction: The Medical Claims application is not available in Datacap FastDoc or Datacap Navigator.

- [Medical Claims application client procedures](#)
When running any Medical Claims job, some task shortcuts are available by default.
- [Processing your own claim form images](#)
You can easily test Medical Claims using your own medical claim form images, instead of the sample images included with the application. After backing up the sample images, you delete them from the input folder and replace them with your own claims.
- [Medical Claims application in Datacap Web Client](#)
You must install and configure Datacap Web (Datacap Web Server and Datacap Web Client) in order to use the Medical Claims web application. You run the Datacap Medical Claims application in your web browser by starting the Datacap Server Service, logging in to the Medical Claims application in the Datacap Web Client, and running the default Datacap Web Client tasks to process pre-scanned images.

Medical Claims application client procedures

When running any Medical Claims job, some task shortcuts are available by default.

About this task

Restriction: The Medical Claims application on the Datacap Web Client includes only the following shortcuts: Web Scan, Inst_Verify, Prof_Verify, and Upload. The Medical Claims application is not available in Datacap FastDoc or Datacap Navigator.

Table 1. Medical Claims default shortcuts

Medical Claims shortcut	Description
Scan	You select the job and run the institutional or professional IScan tasks to scan medical claims forms into Medical Claims.
VScan	You select the job and run the institutional or professional VScan tasks to bring previously scanned medical claims images into Medical Claims.
Web Scan	You select the job, and run the institutional or professional Web Scan tasks to import previously scanned medical claims images into Medical Claims.
Background	Starts the Batch Profiler task that performs many image processing functions including: identification of pages, image cleaning (e.g. despeckling), image rotation (if necessary), and full page character recognition. It also attempts to identify the claim form by comparing it to a library of known forms.
Inst_Verify	Starts the Verify task for institutional claim forms that displays the form images and the data that has been recognized in the Verify window so you can ensure that the data is complete and correct. When a scanned form is not a known claim form, the Verify window allows you to identify the fields and data on it, and to save its fingerprint so it will be recognized the next time.
Prof_Verify	Starts the Verify task for professional claim forms that displays the form images and the data that has been recognized in the Verify window so you can ensure that the data is complete and correct. When a scanned form is not a known claim form, the Verify window allows you to identify the fields and data on it, and to save its fingerprint so it will be recognized the next time.
Fixup	Allows an operator to fix document integrity problems in batches that failed integrity checks.
Upload	Used by remote users (using Datacap Web client) to upload batches to the Datacap Server.

- [Virtually scanning claim form images using Datacap Desktop](#)
You can run the Medical Claims VScan task to import pre-scanned images without using a physical scanner.
- [Scanning physical claim forms using Datacap Desktop](#)
You can run the Medical Claims Scan task to process hard copy claim forms that have not already been scanned.
- [Running the Medical Claims Background task using Datacap Desktop](#)
The Background task performs many image processing functions including: identification of pages, image cleaning (for example, despeckling), image rotation (if necessary), full page character recognition, and more. It also attempts to identify the claim form by comparing it to a library of known forms.
- [Fixing Medical Claims document integrity problems using Datacap Desktop](#)
The Fixup task allows you to fix document integrity problems in batches that failed integrity checks.
- [Verifying scanned claims using Datacap Desktop](#)
The Verify tasks display the form images and the data that has been recognized in the Verify window so

you can ensure that the data is complete and correct. When a scanned form is not a known claim form, the Verify window allows you to identify the fields and data on it, and to save its fingerprint so it is recognized the next time. Medical Claims includes Verify tasks for both Professional and Institutional medical claim forms.

Parent topic: [Medical Claims application](#)

Virtually scanning claim form images using Datacap Desktop

You can run the Medical Claims VScan task to import pre-scanned images without using a physical scanner.

Before you begin

Before you can run Medical Claims, you need to start the Datacap Server Service. You can then log in to the Medical Claims application through Datacap Desktop or through a web browser by using the Datacap Web Client, and run the default tasks to process sample pre-scanned images.

About this task

You can process the sample, pre-scanned medical claim form images included with the Medical Claims application out of the box using a virtual scan without having to connect a real scanner to the workstation.

For information about how to process your own pre-scanned claim form images through Medical Claims, see [Processing your own claim form images](#).

Procedure

1. On the developer workstation from the Start menu, select IBM Datacap Clients > Datacap Desktop.
2. Log in to Datacap Desktop with account credentials that have sufficient rights to run Medical Claims tasks.
3. Select Medical Claims from the Applications menu.
4. Click the VScan shortcut, and select a job. A Task Monitor window opens and a Datacap Rulerunner window opens displaying a progress bar, then they both close. A message box is displayed indicating the task has finished.
5. Click Stop. The message box closes and the Operations pane is redisplayed. When the VScan task finishes successfully, Medical Claims has picked up the images from the Black or Red folder in either the \MClaims\dco_Institutional\images\ or \MClaims\dco_Professional\images\ filepath (depending on your job selection), and created a Datacap batch, which is a collection of document images to be processed. Datacap has assigned a batch ID and created a uniquely-numbered folder with the name of the batch ID in \MClaims\batches. This folder holds all of the document image processing results for this batch of invoices, including a newly-created .xml file.
6. To view information about the batch, in the Operations pane, click the All shortcut to view the Job Monitor.
7. To refresh the information displayed in the Job Monitor window, click the Job Monitor's title bar and press F5. The Job.Task and Status for this batch have been updated and have progressed to the next task in the workflow.

Parent topic: [Medical Claims application client procedures](#)

Scanning physical claim forms using Datacap Desktop

You can run the Medical Claims Scan task to process hard copy claim forms that have not already been scanned.

Procedure

1. On the developer workstation from the Start menu, select IBM Datacap Clients > Datacap Desktop.
2. Log in to Datacap Desktop with account credentials that have sufficient rights to run Medical Claims tasks.
3. Select Medical Claims from the Applications menu.
4. Click the Scan shortcut, and select a job. A confirmation message is displayed when the task is complete.
5. Click Stop. The message box closes and the Operations pane is redisplayed. When the Scan task finishes successfully, a Scan.xml file has been created (containing page types you have assigned) in the batch's uniquely-numbered folder in \MClaims\batches.

Parent topic: [Medical Claims application client procedures](#)

Running the Medical Claims Background task using Datacap Desktop

The Background task performs many image processing functions including: identification of pages, image cleaning (for example, despeckling), image rotation (if necessary), full page character recognition, and more. It also attempts to identify the claim form by comparing it to a library of known forms.

Procedure

1. On the developer workstation from the Start menu, select IBM Datacap Clients > Datacap Desktop.
2. Log in to Datacap Desktop with account credentials that have sufficient rights to run Medical Claims tasks.
3. Select Medical Claims from the Applications menu.
4. Click the Background shortcut. A Task Monitor window opens and a Datacap Rulerunner window opens displaying a progress bar, then they both close. When processing is complete, a message box is displayed indicating the task has finished.
5. Click Stop. The message box closes and the Operations pane is redisplayed. When the Background task finishes successfully, the images have been organized into documents, copies (TIO) have been made of the original images, fingerprint (CCO) and page XML files have been created, along with an .xml file in the batch's uniquely-numbered folder in \MClaims\batches.
6. To view information about the batch, in the Operations pane, click the All shortcut to view the Job Monitor.
7. To refresh the information displayed in the Job Monitor window, click the Job Monitor's title bar and press F5. The Job.Task and Status for this batch have been updated and are now in verify and pending.

Parent topic: [Medical Claims application client procedures](#)

Fixing Medical Claims document integrity problems using Datacap Desktop

The Fixup task allows you to fix document integrity problems in batches that failed integrity checks.

Procedure

1. On the developer workstation from the Start menu, select IBM Datacap Clients > Datacap Desktop.
2. Log in to Datacap Desktop with account credentials that have sufficient rights to run Medical Claims tasks.
3. Select Medical Claims from the Applications menu.
4. Click the Fixup shortcut. A Task Monitor window opens and a Datacap Rulerunner window opens displaying a progress bar, then they both close. A message box is displayed indicating that the Fixup task finished.
5. Click Stop. The message box closes and the Operations pane is redisplayed.
6. To view information about the batch, in the Operations pane, click the All shortcut to view the Job Monitor.
7. To refresh the information displayed in the Job Monitor window, click the Job Monitor's title bar and press F5.

Parent topic: [Medical Claims application client procedures](#)

Verifying scanned claims using Datacap Desktop

The Verify tasks display the form images and the data that has been recognized in the Verify window so you can ensure that the data is complete and correct. When a scanned form is not a known claim form, the Verify window allows you to identify the fields and data on it, and to save its fingerprint so it is recognized the next time. Medical Claims includes Verify tasks for both Professional and Institutional medical claim forms.

Procedure

To verify scanned claims:

1. On the developer workstation from the Start menu, select IBM Datacap Clients > Datacap Desktop.
2. Log in to Datacap Desktop with account credentials that have sufficient rights to run Medical Claims tasks.
3. Select Medical Claims from the Applications menu.
4. Click either the Prof_Verify or Inst_Verify shortcut, depending on the forms to be scanned. A Task Monitor window opens and a Datacap Rulerunner window opens, then the Verify window opens. Important: For demonstration purposes only, all forms are deliberately set to be displayed in the Verify window. In a production environment, you decide whether you want every form to be displayed to the operator who is running the Verify task, or if the operator will only see those forms with recognition results that are low confidence and with data validation rules that have been violated.

The Verify window displays the first claim form in the batch and the details for the first field on that claim form. The window displays fields, recognized data, snippets, and a number of buttons, with form header information displayed on top, and detail on the bottom. The window also displays the scanned form.

When there are errors or low confidence results, the cursor is placed in the first problem field and the field is outlined in red. In addition, the background color of a data field indicates:

- o Blue: No data validation errors or low confidence recognition results
- o Yellow: Low confidence recognition results (low confidence characters are displayed in red)
- o Red: Data validation error

When a field fails validation (red background), you must correct the data before you can complete the batch. When the recognition results for a field are low confidence (yellow background), you can either update the contents of the field or ignore the low confidence condition. Click the Next Edit button (or keyboard shortcut `Alt+I`) to move to the next red or yellow problem field. When all problem fields on a page have been inspected and fixed or ignored, the page can be revalidated by clicking the Run Validations button (or keyboard shortcut `Alt+V`) which will change the background color of the fields from red to blue if the validation succeeds, and from yellow to blue if the field was edited. When done

with the page, click the Next Problem button (or keyboard shortcut **Ctrl+N**) to move to the first problem on the next page.

Table 1. Medical Claims Verify window hot-keys. The following table describes some of the things you can do using buttons and hot-key combinations on the Medical Claims Verify window.

To	Do this
Move to the next data field	Press <Tab>
Move to next low confidence character	Press <Alt+L>
Fill empty field with data using Click N Key	Click in an empty data entry field, then do one of the following: Point at the field value on the document image and click it, or, Click and drag the mouse to draw a rectangular box around the existing field data on the image, or, Press and hold <Alt>, click and drag the mouse to draw a rectangular box that encompasses additional blank space (to leave room for additional characters), or, Press and hold <Shift>, click a field that has already been defined and drag the mouse to extend the field.
Invoke validation rules for entire invoice	Press <Alt+V>
Delete the document that is currently displayed	Press <Alt+D>
Mark the document for review	Press <F2>
Display the next document in the batch	Press <Ctrl+Shift+N>
Display the previous document in the batch.	Press <Ctrl+Shift+P>
Move to the next problem document in the batch	Press <Ctrl+N>
Move to the previous problem document in the batch	Press <Ctrl+P>
Display original version of scanned image (with lines, noise, etc.), display cleaned up version of image (toggles from TIO to TIF)	Click the TIO button to toggle to TIF
Display or hide the scanned image	On the View menu, select Image View, or press <Ctrl+Alt+I>
Move the image within the display frame	Click the image pane header and drag

5. After you have reviewed and validated each sample document, an End of Page Reached message will be displayed.
6. Click Yes to go to the next document. When you have finished processing the last form in the batch, a message box indicates that no additional problems were found.
7. Click OK to finish the batch. A message box is displayed indicating the task has finished.
8. Click Stop. The message box closes and the Operations pane is redisplayed. When the Verify task finishes successfully, page XML files have been updated (for example, when low confidence characters

have been changed during Verify), and a Verify.xml file has been created in the batch's uniquely-numbered folder in \MClaims\batches.

9. To view information about the batch, in the Operations pane, click the All shortcut to view the Job Monitor.
10. To refresh the information displayed in the Job Monitor window, click the Job Monitor's title bar and press F5.

Parent topic: [Medical Claims application client procedures](#)

Processing your own claim form images

You can easily test Medical Claims using your own medical claim form images, instead of the sample images included with the application. After backing up the sample images, you delete them from the input folder and replace them with your own claims.

About this task

This procedure will allow you to test the processing of your own images, on a small scale. In a production environment, you will probably make some changes to the VScan task to handle larger quantities of files, as well as use a network shared folder for the input folder, instead of the pre-configured local application folder.

Procedure

To set up Medical Claims to test scanning your own claim form images:

1. Back up and then delete the sample images. Navigate to the \\Server\Datacap\Medical Claims\images\Input folder in Windows Explorer, and make a backup of the directory. Once you have created the backup, delete the sample images in the original \Input folder.
2. Copy your own invoice images to the \\Server\Datacap\Medical Claims\images\Input folder.

Results

You can now run Medical Claims tasks on your own claim form images, instead of the sample images.

Parent topic: [Medical Claims application](#)

Medical Claims application in Datacap Web Client

You must install and configure Datacap Web (Datacap Web Server and Datacap Web Client) in order to use the Medical Claims web application. You run the Datacap Medical Claims application in your web browser by starting the Datacap Server Service, logging in to the Medical Claims application in the Datacap Web Client, and running the default Datacap Web Client tasks to process pre-scanned images.

About this task

The default shortcuts you run, and the sequence in which you run them using the Medical Claims application in Datacap Web Client are as follows:

Table 1. Medical Claims shortcuts in the Datacap Web Client

Medical Claims	Allows you to
Web Scan	Process sample, pre-scanned images

Medical Claims	Allows you to
Upload	Move scanned images to the Datacap Server Service for further processing
Verify (as Inst_Verify and Prof_Verify)	Validate claim data

Important: To complete processing, the Medical Claims Fixup, Batch Profiler, and Export tasks must be run manually from Datacap Desktop. In a client/server environment, the Medical Claims Batch Profiler and Export tasks can be run automatically as background tasks by using either the Datacap Client Service or Datacap Rulerunner.

- [Logging in to Medical Claims in Datacap Web Client](#)
This procedure provides instructions on how to log in to the Medical Claims application in Datacap Web Client using a web browser.
- [Viewing Medical Claims permissions in Datacap Web Client](#)
When you perform the default Datacap installation and configuration, some Datacap groups, users, passwords, and stations are automatically set up for you in the Medical Claims application. You can review and make changes to these permissions in the Datacap Web Client.
- [Scanning claim forms in Datacap Web Client](#)
This procedure provides instructions on how to run the Medical Claims virtual scan task in the Datacap Web Client and begin processing the sample claim forms.
- [Uploading Medical Claims scanned images in Datacap Web Client](#)
- [Verifying Medical Claims data in Datacap Web Client](#)
This procedure provides instructions on how to run the Medical Claims Verify task in the Datacap Web Client. Depending on what type of claim form you are verifying, you will run either the Inst_Verify or Prof_Verify shortcuts.

Parent topic: [Medical Claims application](#)

Logging in to Medical Claims in Datacap Web Client

This procedure provides instructions on how to log in to the Medical Claims application in Datacap Web Client using a web browser.

Before you begin

Before you can connect to Datacap Web Client, you need to [start the Datacap Server Service](#). You can then log in to the Medical Claims application.

Procedure

1. Open your browser, and enter the address of the Datacap Web Server followed by the alias for the website.
 - o If you installed Datacap on a single machine, enter `http://localhost:port/tmweb.net`.
 - o If you installed Datacap in a client/server environment and you are connecting from a remote workstation, enter the IP address of the Datacap Web Server followed by the alias for the website (for example: `http://127.0.0.1:port/tmweb.net`, `http://WebServerName:port/tmweb.net`, or `https://WebServerName:port/tmweb.net`).

After a pause, the Datacap Login window opens.

Tip: You might have to turn off your browser's popup blocker the first time you display the Login page.

2. On the Datacap Web Client login page, enter the following:

Table 1. Datacap login

page values for Medical Claims

Field	Value
Application	Medical Claims
User ID	admin
Password	admin
Station	1

3. Click Login. The Medical Claims main window opens with the Operations pane and available shortcuts displayed.

Parent topic: [Medical Claims application in Datacap Web Client](#)

Viewing Medical Claims permissions in Datacap Web Client

When you perform the default Datacap installation and configuration, some Datacap groups, users, passwords, and stations are automatically set up for you in the Medical Claims application. You can review and make changes to these permissions in the Datacap Web Client.

Before you begin

Before you can connect to Datacap Web Client, you need to [start the Datacap Server Service](#). You can then log in to the Medical Claims application.

Remember: The default permissions are configured for use with native Datacap authentication. If you are using other authentication methods, you must set up users and groups in advance. See [Planning your Datacap system](#) and [Configuring authentication for Datacap](#).

Procedure

1. After logging in to the Medical Claims application on Datacap Web Client, click on the Administrator tab, and click either Groups, Users, or Stations.
2. Click on any group, user, or station listed. If you are viewing a group, the privileges, permissions, and users for that group are displayed. The options for a user include privileges and permissions, and the options for stations just include the permissions.

Parent topic: [Medical Claims application in Datacap Web Client](#)

Scanning claim forms in Datacap Web Client

This procedure provides instructions on how to run the Medical Claims virtual scan task in the Datacap Web Client and begin processing the sample claim forms.

Before you begin

Before you can connect to Datacap Web Client, you need to [start the Datacap Server Service](#). You can then log in to the Medical Claims application.

Procedure

1. After logging in to the Medical Claims application on Datacap Web Client, click the Web Scan shortcut on the Operations tab. The list of available jobs is displayed:

INST Web Black
 Virtual scan of black institutional claim forms
 INST Web Red
 Virtual scan of red institutional claim forms
 PROF Web Black
 Virtual scan of black professional claim forms
 Prof Web Red
 Virtual scan of red professional claim forms

The Virtual scanning batch page opens. By default, the Multiple files checkbox is filled.

2. The Scanned Image Folder field on the page indicates where the pre-scanned images are temporarily placed. If you do not have permission to write to this location, change this path to one to which you can write.
3. In the Source Directory field, click Browse and go to the folder where the pre-scanned images are installed. The default location is in the application folder within subdirectories for the job you selected:

INST Web Black
 \Datacap\Medical Claims\dco_Institutional\images\Black
 INST Web Red
 \Datacap\Medical Claims\dco_Institutional\images\
 PROF Web Black
 \Datacap\Medical Claims\dco_Professional\images\Black
 Prof Web Red
 \Datacap\Medical Claims\dco_Professional\images

4. Select the first image file in the folder and click Open. The path to the image file is displayed in the Source Directory field.
5. You can change the number in the Expected pages field to the number of images you want to process if you do not want to process all of the sample claim forms. Then, click Scan. The pages are scanned and images of the pages are displayed. The batch-specific name of the folder is appended to the Scanned Image folder path and an End of scan message is displayed.
6. The following table describes some of the things you can do on the Web Demo Scan dialog:

Table 1. Functions that you can perform on the Web Scan dialog

To	Do this step
Change sequence of pages	Select either the image or the name of the image that you want to move. Then, click either the MoveUp or MoveDown arrow
Delete all pages	Click Remove All
Delete a single page	Select the image that you want to delete, then click Remove
Enlarge scanned images	Click Zoom in
Shrink scanned images	Click Zoom out
Rotate an image	Click Rotate 90
Insert the image before the first image in the batch	Click Insert before
Complete the Scan Task processing of the batch	Click Done

To	Do this step
Cancel the Scan Task processing of the batch; deletes batch and all batch data from system	Click Cancel
Put the batch on hold and end processing of the batch; batch remains available to same user	Click Hold

7. After all of the images are scanned, click Done. A message box is displayed indicating the Batch finished with a status of finished.
8. Click OK. A message box asks if you want to continue processing.
9. Click Stop. The message box closes and the Operations pane opens.
10. When the Web Scan task finishes successfully, the source images are collected into a batch. These images are copied to the uniquely numbered folder of the batch in the C:\Datacap\Scan folder, or the folder you specified in the Scanned Image Folder field.
11. Click the Monitor tab to view information about the batch.
12. Press F5 to refresh the information. The Job, Task, and Status for this batch are updated.

Parent topic: [Medical Claims application in Datacap Web Client](#)

Uploading Medical Claims scanned images in Datacap Web Client

Before you begin

Before you can connect to Datacap Web Client, you need to [start the Datacap Server Service](#). You can then log in to the Medical Claims application.

About this task

This procedure provides instructions on how to run the Medical Claims upload task.

Procedure

1. After logging in to the Medical Claims application on Datacap Web Client, click the Upload shortcut on the Operations tab. The Upload window opens and a transferring the batch to server message is displayed briefly, then closes. A message box is displayed indicating the batch has finished with a status of finished.
2. Click OK. A message box asks if you want to continue processing.
3. Click Stop. The message box closes and the Operations pane is redisplayed.
4. When the Upload task finishes successfully, the scanned images have been moved to the batch's uniquely-numbered folder found in \Datacap\Medical Claims\batches and ivscan and upload .xml files have been created.
5. Click the Monitor tab to view information about the batch. To refresh the information displayed, press F5. The Job, Task, and Status for this batch have been updated.

Parent topic: [Medical Claims application in Datacap Web Client](#)

Verifying Medical Claims data in Datacap Web Client

This procedure provides instructions on how to run the Medical Claims Verify task in the Datacap Web Client. Depending on what type of claim form you are verifying, you will run either the Inst_Verify or Prof_Verify shortcuts.

Before you begin

Before you can connect to Datacap Web Client, you need to [start the Datacap Server Service](#). You can then log in to the Medical Claims application.

Procedure

1. To run the Medical Claims Verify task:
2. After logging in to the Medical Claims application on Datacap Web Client, click the appropriate Verify shortcut on the Operations tab. For professional claim forms, use Prof_Verify. For institutional forms, use Inst_Verify. The verification panel opens with the first image displayed.
3. The following table describes some of the things you can do on the verification panel:

Table 1. Medical Claims Verify task options

Option	How to perform this task
Move to the next data field	Press Tab
Move to the next low confidence character	Click Next LC
Fill empty field with data using Click N Key	Click in the empty data entry field, then do one of the following: <ul style="list-style-type: none"> o Click the field value on the image. o Click and drag the mouse to draw a box around the existing field data on the image.
Complete processing of image and display next image	Click Submit
Display the first page of the next document in the batch without submitting or revalidating data	Click Next
Display the first page of the previous document in the batch without submitting or revalidating data	Click Prev
Put entire batch on hold	Click Hold

4. When you have finished with the last document and click Submit, a message box is displayed indicating all documents are complete and you are prompted to finish the batch.
5. Click OK. The message box closes and another message box is displayed indicating the task on the batch has finished with a status of finished.
6. Click OK. A message box asks if you want to continue processing.
7. Click Stop. The message box closes and the Operations pane is redisplayed.
8. When the Verify task finishes successfully, page .xml files have been updated (for example, when low confidence characters have been changed during Verify), and a Verify.xml file has been created in the batch's uniquely-numbered folder found in \Datacap\Medical Claims\batches.
9. Click the Monitor tab to view information about the batch. To refresh the information displayed, press F5. The Job, Task, and Status for this batch have been updated.

Parent topic: [Medical Claims application in Datacap Web Client](#)

TravelDocs application

The TravelDocs sample application that is included in the Datacap installation illustrates how a Datacap application processes various travel-related documents.

The TravelDocs application is delivered with pre-configured jobs and tasks so that you can create batches with the provided pre-scanned sample images. You can create and process batches by using the Datacap Desktop and Datacap web clients without connecting a scanner to your computer. You can then process the captured data to the Export task.

A job is a collection of the tasks that are needed to input, identify, recognize, verify, and export data and images.

Each task accomplishes a portion of the required steps that are needed to capture and ensure that the captured data and images are complete, accurate, and valid.

When you use the Datacap Web Client, you can click Shortcuts on the Operations tab and run various jobs and tasks.

Attention: Ensure that the Datacap Server Service is started before running the Datacap TravelDocs application. See [Ensuring that the Datacap Server Service is started](#).

The following jobs are included in the TravelDocs application.

- Main Job
- FixUp Job
- Web Job

The sequence in which you run the tasks in these jobs is as follows.

Table 1. Sequence for running tasks in the Main Job and Web Job

Task name	Does this
VScan (Datacap Desktop) or Web VScan (Datacap Web Client)	Process sample, pre-scanned images.
Upload (Datacap Web Client only)	Move scanned images to the Datacap Server for further processing.
PageID (Datacap Desktop only)	Identify and assign page type to image.
Batch Profiler (Datacap Desktop only)	Perform recognition and capture data on the page.
FixUp (Datacap Desktop only)	Correct document structure issues.
Verify (Datacap Desktop) or Web Verify (Datacap Web Client)	Perform data validation and correction.
Export (Datacap Desktop)	Complete the processing of images and produce output.

- [Accessing the Datacap Web Client Login page and logging in](#)
You log in to the Datacap Web Client TravelDocs application by using Internet Explorer.
- [Running the TravelDocs VScan task](#)
In Datacap Desktop, you can run the TravelDocs VScan task on the sample images that are included with the TravelDocs application to create a batch.
- [Running the TravelDocs Web VScan task](#)
You can run the TravelDocs Web VScan task on the sample images that are included with the TravelDocs application to create a batch.

- [Running the TravelDocs Web Upload task](#)
You can move scanned images from a remote Datacap client to the Datacap Server by using the Web Upload task.
- [Running the TravelDocs PageID task](#)
In Datacap Desktop, you can run the TravelDocs PageID task manually to confirm that the Page ID task operates properly.
- [Running the TravelDocs Batch Profiler task](#)
In Datacap Desktop, you can run the TravelDocs Batch Profiler task manually to confirm that the task operates properly.
- [Running the TravelDocs Verify task](#)
In Datacap Desktop, you can run the TravelDocs Verify task on the sample images that are included with the TravelDocs application.
- [Running the TravelDocs Web Verify task](#)
You can run the Web Verify task on the sample images that are included with the TravelDocs application.
- [Running the TravelDocs Export task](#)
You can run the TravelDocs Export task manually in Datacap Desktop to confirm that the Export task operates properly.

Accessing the Datacap Web Client Login page and logging in

You log in to the Datacap Web Client TravelDocs application by using Internet Explorer.

Procedure

To accessing the Datacap Web Client Login page and login:

1. Start Internet Explorer.
2. When you run Internet Explorer on the same computer as Datacap Web Client, enter the default web server address `http://localhost` and press Enter. If you are running Datacap Server and you run Internet Explorer from a different computer, enter either the IP address of the web server or name `http://WebServerName`. Then, press Enter. The first time that you access the Login page, there is a pause before the Datacap Web Client Login page is displayed.
3. Select the TravelDocs application, and change the User, Password and Station to `admin`, `admin` and `1`, then click Login.

Parent topic: [TravelDocs application](#)

Running the TravelDocs VScan task

In Datacap Desktop, you can run the TravelDocs VScan task on the sample images that are included with the TravelDocs application to create a batch.

About this task

In a production environment, the images that comprise Datacap batches can come from a number of sources, including scanners, email systems, and fax systems. Batches can be created manually or automatically. You can run the VScan task to create a batch, and confirm that the task operates properly, without connecting a scanner or configuring an email system.

Procedure

Follow this procedure to run the TravelDocs VScan task in Datacap Desktop.

1. From the Start menu, select Datacap Clients > Datacap Desktop.
2. Enter the User, Password and Station to `admin`, `admin` and `1`.
3. Select TravelDocs from the Applications menu.
4. Select VScan from the Task Shortcuts menu.
5. Resize the Main Job.VScan window so that you can see all of the fields and controls in the Image View panel, the controls in the middle panel, and the Batch View panel.
6. Click the button next to the Scan from field, browse to the `\Datacap\TravelDocs\images` folder, and select the `Car1.tif` file and click Open.
7. On the Main Job.VScan window, change the value in the expected field to `3`, then click VScan.
8. When the three images are displayed, click Done.
9. Click OK.
10. Click Cancel. The following actions are done when you run the VScan task.
 - o Copies images from `\Datacap\TravelDocs\images`.
 - o Creates a batch and assigns a batch ID to it.
 - o Creates the `vscan.xml` file.
 - o Creates a unique-numbered folder under `\Datacap\TravelDocs\batches` (In a client-server environment, the folder will be created on the server).
 - The unique-numbered folder contains all of the image-processing results for this batch, including the `vscan.xml` file.
11. Repeat these instructions to create and queue more batches for the Rulerunner Service to process.

Parent topic: [TravelDocs application](#)

Running the TravelDocs Web VScan task

You can run the TravelDocs Web VScan task on the sample images that are included with the TravelDocs application to create a batch.

About this task

In a production environment, the images that comprise Datacap batches can come from a number of sources, including scanners, email systems, and fax systems. Batches can be created manually or automatically. Use the Web VScan task to create a batch, and confirm that the task operates properly, without connecting to a scanner or configuring an email system.

Procedure

1. On the TravelDocs Operations tab, click the Web VScan shortcut.
2. If necessary, resize the Web Job Virtual scanning window so that you can see all of the fields and controls.
3. Click Browse and go to the `\Datacap\TravelDocs\images` folder.
4. Select the `Car1.tif` file and click Open.
5. Change the value in the Expected pages field to `3`, then click Scan.
6. When the three images are displayed, click OK, then click Done.
7. Click OK.
8. Click Stop. Web VScan completes the following tasks.
 - o Copies images from `\Datacap\TravelDocs\images`
 - o Creates a batch and assigns a batch ID to it
 - o Creates a unique-numbered folder in the location that is specified in the Scanned Image Folder field (defaults to the `c:\datacap\scan` folder)
9. Repeat these instructions to create and queue more batches for the Rulerunner Service to process.

Parent topic: [TravelDocs application](#)

Running the TravelDocs Web Upload task

You can move scanned images from a remote Datacap client to the Datacap Server by using the Web Upload task.

About this task

In a production environment, the Web Upload task is automatically run to move scanned images from a remote Datacap client to the Datacap Server. This procedure provides instructions on how to use the Web Upload task manually to confirm that the task functions properly without configuring the Datacap Web Client Upload Service.

Procedure

1. With the TravelDocs Operations > Run Shortcut tab displayed, click the Upload shortcut.
2. When all of the images from the batch are uploaded, click OK.
3. Click Stop. When the task is successfully finished, the images have been moved from the uniquely-numbered folder in the folder specified in the Scanned Image Folder field (defaults to the c:\datacap\scan folder) to the folder with the same name in the \Datacap\TravelDocs\batches folder (In a client-server environment, the folder is located on the server). This folder holds all of the image processing results for this batch, including the newly-created vscan.xml file.
4. When running this task to queue up batches for the Rulerunner Service to process, repeat these instructions to upload additional batches.

Parent topic: [TravelDocs application](#)

Running the TravelDocs PageID task

In Datacap Desktop, you can run the TravelDocs PageID task manually to confirm that the Page ID task operates properly.

About this task

In a production environment, Datacap Page Identification tasks are usually run in the background by the Rulerunner Service. You can run the PageID task manually without setting up and configuring Rulerunner Service.

Procedure

1. From the Start menu, select Datacap Clients > Datacap Desktop.
2. Log in using the same default User, Password, and Station (admin, admin, and 1).
3. Select TravelDocs from the Applications menu.
4. Select PageID from the Task Shortcuts menu. The PageID task organizes the images into documents, and saves the following files in the \Datacap\TravelDocs\batches*batch number* folder (In a client-server environment, the folder is located on the server).
 - o Original images as .TIO files
 - o pageid_rrs.log
 - o Fingerprint (CCO) files
 - o PageID.xml

Parent topic: [TravelDocs application](#)

Running the TravelDocs Batch Profiler task

In Datacap Desktop, you can run the TravelDocs Batch Profiler task manually to confirm that the task operates properly.

About this task

In a production environment, Datacap recognition and prevalidation tasks, such as the Batch Profiler task, are usually run in the background by the Rulerunner Service. You can run the Batch Profiler task manually without setting up and configuring Rulerunner Service.

Procedure

1. From the Start menu, select Datacap Clients > Datacap Desktop.
2. Log in using the same default User, Password, and Station (`admin`, `admin`, and `1`).
3. Select TravelDocs from the Applications menu.
4. Select Batch Profiler from the Task Shortcuts menu. The Batch Profiler task captures data from each image, and saves the following files in the `\Datacap\TravelDocs\batches\batch number` folder (In a client-server environment, the folder is located on the server).
 - o For each document, page XML files that contain character confidence values and field position information
 - o `batch profiler_rrs.log`
 - o `Batch Profiler.xml`

Parent topic: [TravelDocs application](#)

Running the TravelDocs Verify task

In Datacap Desktop, you can run the TravelDocs Verify task on the sample images that are included with the TravelDocs application.

About this task

In a production environment, Datacap verification tasks are run manually to confirm that the captured data is correct and complete. Batches can be verified by using either Datacap Desktop or the Datacap Web Client. Follow this procedure to run the TravelDocs Verify task in Datacap Desktop.

Procedure

1. From the Start menu, select Datacap Clients > Datacap Desktop.
2. Change User, Password and Station to `admin`, `admin` and `1`.
3. Select TravelDocs from the Applications menu.
4. Select Verify from the Task Shortcuts menu.
5. Reposition and resize the Main Job.Verify window so that you can see all of the fields and controls in the Image View panel, the controls in the middle panel, and the Batch View panel. Datacap Desktop displays the first document in the batch and the recognized data in the middle pane. Fields that are highlighted in yellow indicate that the recognition results are of low confidence. Fields that are highlighted in red indicate that the results failed to pass validation.

The recognition results for the Pickup Location, Return Location, and Fuel service are of low confidence, but are accurate.

6. Click Submit.
7. Because the recognition results for the Collision Damage Waiver (CDW) check box are of low confidence and inaccurate, change the value in the field from `blank` to `Selected` and click Submit.
8. Because the recognition for the Car Type field did not pass validation, click the Car Type link to display the list of values that you can use in this field.
9. Double-click a value from the list, and click Submit. The Verify saves the following files in the `\Datacap\TravelDocs\batches\batch number` folder (In a client-server environment, the folder is located on the server).
 - o Updated individual page XML files (when data is corrected or low confidence characters are changed)
 - o `verify_rrs.log`
 - o `Verify.xml`

Parent topic: [TravelDocs application](#)

Running the TravelDocs Web Verify task

You can run the Web Verify task on the sample images that are included with the TravelDocs application.

About this task

In a production environment, Datacap verification tasks are run manually to confirm that the captured data is correct and complete. You can verify batches by using either Datacap Desktop or the Datacap Web Client.

Procedure

1. In the TravelDocs Operations tab, click Verify, and resize the Verify window so that you can see as many of the fields and controls as possible. Datacap displays the first document in the first batch, with the recognized data displayed in the middle pane. Fields that are highlighted in yellow indicate that the recognition results are of low confidence. Fields that are highlighted in red indicate that the results failed to pass validation.

The recognition results for the Pickup Location, Return Location, and Fuel service are of low confidence, but the recognition results are accurate.

2. Click Submit.
3. Because the recognition results for the Collision Damage Waiver (CDW) check box are of low confidence and inaccurate, change the value in the field from `blank` to `Selected` and click Submit.
4. Because the recognition for the Car Type field did not pass validation, click the Car Type link to display a list of values that you can use in this field.
5. Double-click a value from the list, and click Submit.
6. Click OK.
7. Click OK.
8. Click Stop. The Verify saves the following files in the `\Datacap\TravelDocs\batches\batch number` folder (In a client-server environment, the folder is located on the server).
 - o Updated individual page XML files (when data is corrected or low confidence characters are changed)
 - o `verify_rrs.log`
 - o `Verify.xml`

Parent topic: [TravelDocs application](#)

Running the TravelDocs Export task

You can run the TravelDocs Export task manually in Datacap Desktop to confirm that the Export task operates properly.

About this task

In a production environment, Datacap export tasks are usually run unattended in the background by the Rulerunner Service. You can run the TravelDocs Export task manually without setting up and configuring Rulerunner Service.

Procedure

1. From the Start menu, select Datacap Clients > Datacap Desktop.
2. Log in using the same default User, Password, and Station (`admin`, `admin`, and `1`).
3. Select TravelDocs from the Applications menu.
4. Select Export from the Task Shortcuts menu. The Export task updates the TravelDocs export database in `\Datacap\TravelDocs\TravelDocsExport.mdb` and saves the following files in the `\Datacap\TravelDocs\batches\batch number` folder (In a client-server environment, the folder is located on the server).
 - o `export_rrs.log`
 - o `Export.xml`

Parent topic: [TravelDocs application](#)

Configuring Rulerunner to run application tasks

Once you've successfully configured your applications to run tasks manually, you can configure Rulerunner to run the application's background tasks automatically. Some of the tasks that you will want to configure Rulerunner to run include recognition, image pre-processing, validation, and export tasks. Virtual scan tasks (VScan) can also be run if they are designed to pick up images automatically, rather than requiring a user to select images manually.

You configure Rulerunner to run your application tasks on one or more threads. For example, if you are using the Accounts Payable application, you may want to set Rulerunner up to run the Batch Profiler and Export tasks on either a single thread or run each of them on their own threads. Repeat the instructions as many times as required to set up tasks from multiple applications.

Setting up Rulerunner requires you to:

- Install the Rulerunner service, and set up accounts, security, and permissions to your Datacap and Rulerunner servers, as well as to your application. See [Installing and configuring the Rulerunner Service](#).
- Configure Rulerunner to run tasks on one or more threads. See [Configuring Rulerunner to run your applications](#).

Konfigurowanie czynności skanowania tak, aby puste strony były pomijane

W programie Datacap Navigator można skonfigurować czynność skanowania tak, aby puste strony były pomijane (jeśli skaner obsługuje tę funkcję). Opcja pomijania pustych stron jest dostępna w zaawansowanych ustawieniach skanera. Po skonfigurowaniu tej opcji ustawienie jest zachowywane.

W programie Datacap Navigator wykonaj następujące działania:

1. Dodaj wiersz kodu zaznaczony w następującym fragmencie kodu jako ostatni element potomny w akapicie ScannerSet w kodzie XML czynności skanowania aplikacji.

Uwaga: Dla aplikacji „out-of-the-box” (OOTB) *TravelDocs* należy zaktualizować plik nscan.set.xml.

```
<V label="Enable Autofeeder" n="Autofeed" tip="Controls scanner source
- manual vs. autofeeder option" type="checkbox">1</V>
<V label="Use Duplex Mode" n="Duplex" tip="When enabled - scans both
sides of the image - duplex mode" type="checkbox">0</V>
<V label="Pixel Type" n="PixelFormat" tip="Specify pixel type for scanning
(0-BnW, 1-Grayscale, 2-RGB)" type="text">0</V>
<V label="Bit Depth" n="Bits" tip="Specify bitdepth for above image type"
type="text">1</V>
<V label="Image Resolution DPI" n="Resolution" tip="Specify image resolution
(X and Y resolution assumed the same) DPI" type="text">200</V>
<V label="Paper Size" n="PaperSize" tip="Specify paper size for scanning
(0-none, 1-A4, 2-JISB5)" type="text">0</V>
<V label="Image format" n="Extension"
tip="Specify extension for the scanned images: tif, jpeg, bmp, png"
type="text">tif</V>
<V label="Skip Blank Pages" n="SkipBlankPages"
tip="Specify whether to discard blank pages, default is do not discard.
For some scanners, duplex must be unchecked for this to be enabled."
type="checkbox">0</V>
```

2. Przed aktualizacją pliku DatacapWebPlugin.jar wykona jego kopię zapasową na wypadek konieczności przywrócenia poprzedniej wersji w przyszłości.
3. Zatrzymaj usługę IBM WebSphere Application Server.
4. Zamień zaktualizowany plik DatacapWebPlugin.jar w folderze instalacyjnym programu Datacap.
Uwaga: Przykład: C:\Datacap\tmweb.java
5. Uruchom usługę IBM WebSphere Application Server.
6. Zaloguj się do panelu administratora.
Uwaga: Przykładowy adres URL panelu administratora: http://localhost:9080/navigator/?desktop=admin
7. Kliknij pozycję wtyczki i wybierz program Datacap Navigator. Kliknij opcję Edytuj, a następnie opcję Wczytaj. Następnie kliknij opcję Zapisz i zamknij. Potem kliknij opcję Zamknij, aby wyjść, i wyloguj się z aplikacji.

Aby wypróbować funkcję pomijania pustych stron, wykonaj następujące działania:

1. Zaloguj się do pulpitu aplikacji Datacap.
2. Aby zaktualizować wartość opcji pomijania pustych stron, przejdź do aplikacji NScan.
3. Otwórz kartę Advanced. W sekcji Scanner Setup zaznacz lub odznacz pole wyboru Skip Blank Pages.
4. Zapisz ustawienia. Uruchom czynność skanowania, aby sprawdzić, czy puste strony są pomijane podczas skanowania.

Uwaga: Jeśli chcesz wyłączyć tę funkcję w programie Datacap Navigator, wykonaj kroki 3-7 na zapisanej kopii zapasowej pliku DatacapWebPlugin.jar.

Administering

Administering your Datacap system includes configuring access to Datacap from mobile devices, setting group and user permissions, managing workflow tasks, and configuring shortcuts in your web client. You can monitor system performance, schedule your web client to recycle daily, manage FastDoc files, and maintain fingerprints.

- [Datacap web clients administration](#)

In the Datacap Web Client and the Datacap Navigator web client you can add users, groups, and

stations. Workflows, jobs, and tasks can be added and configured in these web clients. You can also configure and run shortcuts that are mapped to one or more tasks.

- [Datacap Desktop administration](#)
You can configure the application tasks that Datacap Desktop runs and modify the way that those tasks are performed.
- [Datacap Application Copy Tool](#)
The Datacap Application Copy Tool is a migration tool that you can use to copy and move a Datacap application. For example, you can move the application from a test environment to replace an existing application in a production environment.
- [Monitoring system performance with IBM System Dashboard for Enterprise Content Management](#)
You can monitor Datacap system performance through the IBM® System Dashboard for Enterprise Content Management.
- [Shutting down Datacap for maintenance](#)
You must stop the Datacap software applications to perform system-wide maintenance, back up your environment, install new Datacap software, or upgrade or remove existing Datacap software.
- [FastDoc maintenance](#)
FastDoc provides maintenance features that manage the input files, folders, and files that are associated with completed and unfinished batches that can accumulate over time.
- [Maintaining fingerprints by using the Fingerprint Maintenance Tool](#)
Use the Fingerprint Maintenance Tool (FMT) to manage your fingerprints and synchronize information between the Fingerprint database, the Document Hierarchy, and Fingerprint XML data files.

Datacap web clients administration

In the Datacap Web Client and the Datacap Navigator web client you can add users, groups, and stations. Workflows, jobs, and tasks can be added and configured in these web clients. You can also configure and run shortcuts that are mapped to one or more tasks.

In Datacap Navigator, you can use the Classify option to edit page types when they are incorrectly identified. You can also modify the Datacap Navigator layout and other options by changing the user settings.

In both the Datacap Navigator and Datacap Web Client Job Monitor, you can monitor and manage all batches and jobs. You can also run tasks from the Job Monitor.

- [Users, groups, and stations administration](#)
You can assign privileges and permissions to users, groups, and stations to an application by using the Datacap Web Client or Datacap Navigator web client.
- [Workflows, jobs, and tasks administration](#)
You configure workflows, jobs, and tasks in the Datacap Web Client or Datacap Navigator web client.
- [Shortcuts administration](#)
You can configure shortcuts in the Datacap Web Client or Datacap Navigator to run tasks, such as Upload, FixUp, and Verify. The Shortcuts in the web client are mapped to one or more tasks in the workflow.
- [Batch queuing to specific users and stations](#)
You can set up a batch workflow task to allow a user or station to queue the next task to another user or station. Many combinations of users and stations can be set up in your web client to open a batch and queue the next task to another user or station.
- [Effects of application pool recycling on Datacap Web Client batches](#)
The IIS application pool for Datacap Web Client must be scheduled to recycle daily to ensure optimal system performance. If batches are processing when the application pool is recycled, Datacap Web Client attempts to change the status of the batches.
- [Creating a custom column in the Job Monitor](#)
You can define a custom column in the Job Monitor by adding a column in a Datacap Engine database

table.

- [Administering Datacap Navigator](#)

Access the Job Monitor, run tasks, add users and groups, and configure stations and workflows in Datacap Navigator.

Parent topic: [Administering](#)

Users, groups, and stations administration

You can assign privileges and permissions to users, groups, and stations to an application by using the Datacap Web Client or Datacap Navigator web client.

In an environment with multiple Datacap servers, the privileges and permissions assigned to users, groups, and stations for an application are effective only on the Datacap server where the change is made. All Datacap Server Services that connect to the application must be restarted. The server does not need to be restarted but the server service must be restarted.

When Workflow changes are made, all clients should be stopped before hand and then after the Workflow changes all servers recycled before clients are allowed to connect.

- Add a user - You can add a user to your application and assign privileges and permissions to the user. You can also add a user to a Datacap group. When you add a user to a group, the user inherits the privileges and permissions of that group.
- Add a group - You can add a group to your Datacap application and assign privileges and permissions to the group. All users that are members of the group inherit the privileges and permissions that are assigned to the group.
- Add a station - You can add a station to your Datacap application and assign permissions to run specific job tasks from that station.

While creating a Station Datacap Navigator administration view, the user logs in to provide a Station ID. Under the General tab, the value in the Maximum field for maximum sessions for that particular Station must not be greater than "9999".

If the maximum sessions value for any Station is greater than "9999", the user is not able to connect to Datacap Navigator afterward.

- Set virtual stations - You can allow multiple users to log in to a station by setting the maximum number of virtual stations to greater than zero. When a station ID is set up with a Maximum number of virtual stations that is greater than zero, multiple users can log in to the station ID. When you set the maximum number of virtual stations to zero, multiple users are denied authorization to log in to that station.

When you set up virtual stations, Datacap assigns a unique substation identifier to each login and allows multiple logins up to the Maximum number specified. If a user's session times outs and the Maximum number of virtual stations is greater than zero, the user can log in again without requiring support from the system administrator. When the Maximum number of virtual stations is set to zero or the maximum number of virtual stations is reached, the user's next attempt to log in will fail. The system administrator must clear the virtual stations for that station ID to allow the user to log in again.

Parent topic: [Datacap web clients administration](#)

Workflows, jobs, and tasks administration

You configure workflows, jobs, and tasks in the Datacap Web Client or Datacap Navigator web client.

- Workflow - You can create a new workflow and configure the workflow details, including the Name, Description, and Program name. A workflow contains jobs and tasks.
- Job - You can create a new job and configure the job details, including the Name, Description, and Priority. Examples of the different types of jobs are a Main Job, FixUp Job, and Web Job. A job contains tasks, such as Upload, Verify, and Export.
- Task - You can create a new task and configure the task details, including Name, Description, Mode, Queue by, and Store. For more information about the Queue by and Store options, see *Batch queuing to specific users and stations*. To configure the task, select a Program value such as Rulerunner, Datacap Desktop, or Multiple to use with the task. Complete the task configuration with the Setup options, such as Batch Processing, Page Processing, Custom Web Panels, Rulerunner, FastDoc, and other settings.
- [The Job Monitor and workflow administration](#)
Managing the workflow includes running batches, monitoring the job queue, and configuring applications, all of which you can do from the Datacap Web Client or the Datacap Navigator web client. You can monitor the status of batches and run tasks from the Job Monitor.

Parent topic: [Datacap web clients administration](#)

Related concepts:

[Batch queuing to specific users and stations](#)

The Job Monitor and workflow administration

Managing the workflow includes running batches, monitoring the job queue, and configuring applications, all of which you can do from the Datacap Web Client or the Datacap Navigator web client. You can monitor the status of batches and run tasks from the Job Monitor.

During the data capture process, documents go through a workflow that consists of several discrete tasks such as scan, upload, page identification, recognition, validation, verification, and export. Datacap uses a queuing mechanism to move batches of documents through the workflow.

In the Job Monitor, you can view the status of all batches. Each row in the Job Monitor represents one batch. For each batch, you can see its current position in the workflow. For example, a batch that completed the scan and upload tasks is now ready for page identification. In this example, the *Task* is PageID and the *Status* is Pending. A batch that completed page identification and is ready to go through the stage that includes document creation, recognition, and validation. The *Task* is Profile and the *Status* is Pending.

You can run a web-based task on a batch by clicking the QID number in the Job Monitor. If the task is not enabled for your web client and is a Rulerunner or Datacap Desktop task, it cannot be started from the Job Monitor. The QID is displayed but it is not an executable link. Typically, an operator selects the web-based task through the Run Shortcut option, such as Verify or Fixup, and Datacap runs the first batch that is queued for that task. This design allows multiple operators to work from the same job queue and Datacap delivers batches to the operators on demand.

Some tasks do not require an operator. For example, page identification, recognition, and validation are background tasks that run without operator intervention. You can configure Rulerunner to run background tasks automatically. Rulerunner monitors the job queue for batches that are pending for specific background tasks. When a batch is ready, Rulerunner processes it automatically.

Parent topic: [Workflows, jobs, and tasks administration](#)

Shortcuts administration

You can configure shortcuts in the Datacap Web Client or Datacap Navigator to run tasks, such as Upload, FixUp, and Verify. The Shortcuts in the web client are mapped to one or more tasks in the workflow.

In the web client, you configure the shortcut details, including the shortcut Name, Description, Mode, and Permissions. The Mode option determines the behavior of Datacap when a user clicks the shortcut. The following are the Mode options.

- Prompt/Web select: Datacap opens the highest priority-pending job in the queue.
- Auto: Same as Prompt/Web select.
- Manual: Datacap displays the job queue so that the operator can select a batch that is pending or on hold.
- Manual for Hold: If there are batches on hold, Datacap displays the job queue with the jobs that are on hold. If there are no jobs on hold, Datacap opens the highest priority-pending batch in the queue.

Restriction: Shortcuts in Datacap Navigator are not supported for a task when `Multiple` is specified for the Program task setting.

Parent topic: [Datacap web clients administration](#)

Batch queuing to specific users and stations

You can set up a batch workflow task to allow a user or station to queue the next task to another user or station. Many combinations of users and stations can be set up in your web client to open a batch and queue the next task to another user or station.

You can define the users and stations that can open and process a batch through a task. Setting up queuing is optional. You must set up the Store field on a prior task that is performed by the target user or station, so it is available for use by a subsequent task. For example, if the goal is for the user who scans batches to verify all of the batches that they scanned, then set up Store User ID on the scan task and Queue by User on the verify task. Each time a task runs that is configured to Store user or station information, any previously stored user or station information for the batch is overwritten.

The Workflow tab provides you with the ability to change these settings. Changing the settings does not affect batches that are already queued.

The task's Queue by field lets you identify the combination of user ID and station ID that can open a batch that is queued for this task. The options for the Queue by field are:

- None: Any user on any station can open the batch (default setting). If a batch is assigned to a user or station in the Job Monitor, then the Queue by None setting for that batch is overridden and only the assigned user or station can access the batch.
- Station: Only the station that stored the batch can open the batch.
- User: Only the user that stored the batch can open the batch.
- Other Station: The station that stored the batch cannot open the batch.
- Other User: The user that stored the batch cannot open the batch.
- Station and User: Only the same station and the same user that stored the batch can open the batch.
- Station and Other User: Only a different user on the station that stored the batch can open the batch.
- User and Other Station: Only the same user on a different station can open the batch.
- Other Station and Other User: Only a different user on a different station can open the batch.

Ensure that the option you choose for the Store field provides the information that is required by the task that occurs later in the workflow. The options for the Store field are:

- None: No user ID or station ID is stored (default setting).
- Station ID: Stores the station ID with the batch.
- User ID: Stores the user ID with the batch.

- Station ID and User ID: Stores the station ID and the user ID with the batch.

Example 1

The following table illustrates a workflow that is composed of four tasks where the Scan task can be run by any user on any station, and the batch returns to the user who scanned it for verification.

Task	Queue By	Store
Task 1 Scan	None	User ID
Task 2 Background (PageID, Recognition)	None	None
Task 3 Verify	User	None
Task 4 Export	None	None

Example 2

The following table illustrates a two-pass workflow that is comprised of five tasks and each batch is verified twice. These settings ensure that when the first Verify operator finishes with a batch, a different operator performs the second Verify task.

Task	Queue By	Store
Task 1 Scan	None	None
Task 2 Profiler (PageID, Recognition)	None	None
Task 3 Verify1	None	User ID
Task 4 Verify2	Other User	None
Task 5 Export	None	None

Parent topic: [Datacap web clients administration](#)

Effects of application pool recycling on Datacap Web Client batches

The IIS application pool for Datacap Web Client must be scheduled to recycle daily to ensure optimal system performance. If batches are processing when the application pool is recycled, Datacap Web Client attempts to change the status of the batches.

For optimal system performance, the Microsoft Internet Information Services (IIS) Datacap Web Client application pool must be recycled regularly. When you create the Datacap Web Client site, the Datacap Web Client Server Configuration tool sets the tmweb.net AppPool application pool values to recycle daily at 03:00:00 AM. You can change the App Pool Recycle Schedule in the Datacap Web Client Server Configuration tool.

If the application pool recycles while the Datacap Web Client users are processing batches, the browser sessions are closed. Datacap Web Client attempts to change the status of the batches to `Hold`.

Some batches might remain in a running state and can be identified and reset manually in the Datacap Web Client Job Monitor. You can also create a custom Datacap Maintenance Manager application to search for batches that require intervention.

Parent topic: [Datacap web clients administration](#)

Creating a custom column in the Job Monitor

You can define a custom column in the Job Monitor by adding a column in a Datacap Engine database table.

About this task

To create a custom column in the Job Monitor, you must add a column in the `tmBatch` table of the Datacap Engine database. The column name must be preceded by `pb_`, such as `pb_allowed`. Configure the data type for MSSQL as `nvarchar`, for Oracle the data type is `nvarchar2`, and the data type for Access is `text`.

You can use Microsoft Access to define a custom column in a development environment but Access is not supported for group filtering because it cannot perform the necessary query for additive groups.

Procedure

1. In the `tmBatch` table in the Datacap Engine database, add a column with the name preceded by `pb_`, such as `pb_allowed`.
2. Save the table and restart the Datacap server.
3. For SQL only, you must re-create the JobMonitor view by following this procedure.
 - a. Open the JMView view for design.
 - b. Delete each item that appears in the *Alias* column.
 - c. Press the Exclamation Mark icon.
 - d. Save the view.
 - e. Open the JobMonitor view for design.
 - f. Delete each item that appears in the *Alias* column.
 - g. Press the Exclamation Mark icon.
 - h. Save the view.

Parent topic: [Datacap web clients administration](#)

Administering Datacap Navigator

Access the Job Monitor, run tasks, add users and groups, and configure stations and workflows in Datacap Navigator.

About this task

- [Datacap Navigator user settings](#)
Configure the layout and behavior of Datacap Navigator by changing your user settings. To change the user settings from any Datacap Navigator view, click the user ID drop-down menu and select Change User Settings.
- [Administration view](#)
As an administrator, you can configure your application and the application's components, and set or modify user permissions and privileges by using the Datacap Navigator Administration view.
- [Batch status \(Job Monitor\)](#)
You can monitor or change the status of all batches and jobs and run tasks from the Datacap Navigator Job Monitor page.
- [Speed Scan and Speed Index](#)
The Speed Scan and Index in IBM Datacap Navigator provides high-speed scanning and indexing scenarios for better performance, as well as enhanced accessibility and usability features, and thus simplifies the end-user actions.

- [Image scans](#)
You can scan your source images and collect them into batches by using Datacap Navigator. To open the scan view, click the scan shortcut in the Datacap Navigator client.
- [Batch verification](#)
You run the Verify task in Datacap Navigator to ensure that your data was accurately captured and recognized by your application. In a production environment, Datacap verification tasks are run manually to identify and correct potential data problems before you upload the images to Datacap.
- [Batch upload](#)
After you run the Scan task to capture your batch, you run the Upload task to upload the images to the Datacap server. While this is a task that is automated by Rulerunner in a production environment, you can upload images manually in Datacap Navigator.
- [Page classification](#)
Document assembly is usually an automatic process that runs in the background. However, in some cases the automatic assembly task does not correctly identify page types. When page types are incorrectly identified, you can use Classify to manually edit the type for each page and modify the batch structure.
- [Batch processing \(Task List\)](#)
You can monitor the status of batches and jobs for a specific task and run pending jobs from the Datacap Navigator Task List page.
- [Setting Datacap Navigator default page layouts](#)
The Classify, Verify, and Scan pages contain widgets such as the image viewer, start panel, field details, and batch structure. You can set the default location of the widgets for specific tasks.
- [Constructing a URL for Datacap Navigator](#)
You can construct a URL that an external application can use to access Datacap Navigator features and actions. You can append parameters to the URL that run actions and open Datacap Navigator pages.
- [External data services for Datacap Navigator](#)
You can use the IBM® Content Navigator external data services (EDS) REST protocol to get data such as files or database tables from an external source. You can use the external data to customize field properties and manage property behavior in Datacap Navigator.
- [Customizing Job Monitor](#)
You can customize the Datacap Navigator Job Monitor page by using external data services (EDS). For example, you can change cell values, styles, column names, and enable cells to show Dojo widgets.
- [Datacap Navigator access](#)
You can access Datacap Navigator in an administrator view or a non-administrator view.

Parent topic: [Datacap web clients administration](#)

Datacap Navigator user settings

Configure the layout and behavior of Datacap Navigator by changing your user settings. To change the user settings from any Datacap Navigator view, click the user ID drop-down menu and select Change User Settings.

For example, you can configure the following layouts and behaviors:

- Specify the station ID that you want to log in to.
- Arrange the widgets for the scan task page.
- Automatically submit a batch after the upload task is completed.
- Automatically override a validation failure during verification.

Parent topic: [Administering Datacap Navigator](#)

Administration view

As an administrator, you can configure your application and the application's components, and set or modify user permissions and privileges by using the Datacap Navigator Administration view.

To open the Administration view, click the Open Datacap Administration View icon in the Datacap Navigator client.

Parent topic: [Administering Datacap Navigator](#)

Related concepts:

[Administering](#)

Related information:

[Creating jobs and tasks for your workflow](#)

[Workflows, jobs, and tasks](#)

[Planning stations for your Datacap applications](#)

Batch status (Job Monitor)

You can monitor or change the status of all batches and jobs and run tasks from the Datacap Navigator Job Monitor page.

During the data capture process, documents go through a workflow that consists of several discrete tasks such as scanning, upload, page identification, recognition, validation, verification, and export. Datacap uses a queuing mechanism to move batches of documents through the workflow.

You can run a task on a batch by clicking the batch in the Job Monitor page and selecting one of the available actions. If the task is not enabled for Datacap Navigator and is a Rulerunner or Datacap Desktop task, it cannot be started from the Job Monitor page.

To find a batch more quickly, enter information from any column in the Batch Filter field. For example, enter `pending` in the Batch Filter field to view only the jobs with a pending status.

Note: On the Job Monitor page, you can use the search filters to limit the batches that are displayed for an application; for example, TravelDocs. The values that you specify in the search filters for each application are saved in a browser cookie. When you access the application next time by using the same browser on the same computer, the same search filters are reapplied. The saved search filters are removed from the cookie when you clear the browser cache or when you do not use Datacap Navigator for a month.

To customize the columns of the table, click the user ID list menu and select Change User Settings and click the Job Monitor tab.

Tip: Click Refresh to update the Job Monitor frequently because other operators might be changing the status of batches or jobs.

- [Batch structure labels](#)
The batch structure shows the labels for the batches instead of document or page IDs.

Parent topic: [Administering Datacap Navigator](#)

Batch structure labels

The batch structure shows the labels for the batches instead of document or page IDs.

When an application has labels, enabled with globalization, set for its application defined by the following two keys in the application resources.json, the batch structure for the Scan, Fixup, and Verify clients shows the appropriate labels for particular batches:

- `"dco.DocumentLabel": "Document"`,
- `"dco.PageLabel": "Page"`,

For documents, the label of a Document *i* is used where "i" is the index. For pages, the label of Page *n* is used, where "n" is the page number.

This feature is applicable in all cases, whether at initialization of client or during editing as in Fix up or indexing where new documents and pages might be created at run time by using some action. When labels are used for display, and you move mouse pointer over the DCO element (document or page), the hover text shows the ID for the document or page. If none of those new JSON properties are defined, the processing defaults to the use of ID for documents and pages in the batch structure.

Parent topic: [Batch status \(Job Monitor\)](#)

Klienty Speed Scan i Speed Index

Klienty skanowania szybkiego (Speed Scan) i indeksowania szybkiego (Speed Index) w programie Datacap Navigator umożliwiają szybkie skanowanie i indeksowanie, przyczyniając się do wzrostu wydajności oraz ułatwienia w obsłudze i dostępie do funkcji, a także upraszczają operacje wykonywane przez użytkowników końcowych.

Klienty Speed Scan i Speed Index

Klienty Speed Scan i Speed Index oferują wyższą wydajność oraz ułatwienia w obsłudze i dostępie do funkcji, a także upraszcza operacje wykonywane przez użytkowników końcowych.

Cechy wyróżniające klientów Speed Scan i Speed Index:

Większa wydajność

W programie Datacap Navigator dostępne są następujące funkcje i zmiany.

- W kliencie Speed Scan jako opcja domyślna dostępne jest przesyłanie asynchroniczne. Użytkownicy nie muszą czekać z rozpoczęciem następnej czynności na zakończenie przesyłania. Zadania w kolejce przesyłania asynchronicznego są automatycznie wznowiane po błędach, gdy tylko użytkownik z powrotem zaloguje się do aplikacji.
- Zwiększono wydajność poprzez skrócenie czasu potrzebnego na skanowanie i wyświetlanie stron w aplikacji Datacap.

Udoskonalona obsługa przeglądarki dokumentów Daeja ViewONE Virtual

Implementacja przeglądarki dokumentów Daeja ViewONE Virtual w programie Datacap Navigator przynosi następujące korzyści:

- Ulepszony widok miniatur w kliencie Speed Scan Client umożliwia zmianę powiększenia miniatur.
- Ulepszony sposób przenoszenia aktywności za pomocą klawisza Tab w przeglądarce.
- Zachowywanie widoku i rozmiaru miniatur po odłączeniu i przyłączeniu okna przeglądarki.

Udoskonalenia projektanta paneli

Następujące funkcje są dostępne w programie Datacap Navigator, a administrator oprogramowania Datacap może wykonywać następujące działania:

- Definiowanie przycisku wyszukiwania i odwzorowanie profilu czynności zdefiniowanego w Datacap Studio na potrzeby wyszukiwania za pośrednictwem usług REST/SOAP/bazy danych.
- Możliwość skonfigurowania kolejności przechodzenia klawiszem Tab i skrótu klawiszowego przycisku wyszukiwania.

Ułatwienia w obsłudze

W programie Datacap Navigator dostępne są następujące funkcje i zmiany:

- Skrócenie sekwencji przechodzenia klawiszem Tab między strukturą partii a panelem pól w kliencie indeksowania – dla usprawnienia pracy.
- Dodany przycisk Wyświetl historię w klientach Speed Scan i Speed Index, który ułatwia odwoływanie się do historii partii.
- Dodano okno dialogowe potwierdzenia wyświetlane, gdy użytkownik zamyka klienta.

Centralne ustawienia skanowania

W programie Datacap Navigator dostępne są następujące funkcje i zmiany:

- Administrator oprogramowania Datacap może teraz zdefiniować grupę użytkowników uprawnionych do modyfikowania ustawień skanera w kliencie Speed Scan.
- Administrator oprogramowania Datacap może określić, które ustawienia skanera będą widoczne w czynności szybkiego skanowania.
- Administrator oprogramowania Datacap może również określić, które ustawienia skanera będą edytowalne w czynności szybkiego skanowania.
- Obsługa większego formatu papieru.

Konfigurowalne klawisze skrótów

Korzystając z konsoli administracyjnej programu IBM Content Navigator, administrator oprogramowania Datacap może przypisywać klawisze skrótów do przycisków menu zdefiniowanych w aplikacji Datacap.

- [Konfigurowanie programu Datacap Navigator i paneli niestandardowych](#)
Datacap Navigator jest klientem WWW programu opartym na programie IBM® Content Navigator. Klienta Datacap Navigator można skonfigurować, ładując wtyczkę i konfigurując repozytoria oraz pulpity w programie IBM Content Navigator.
- [Klient Batch Preparation](#)
Batch Preparation umożliwia przygotowywanie stron lub partii do szybkiego skanowania.
- [Tworzenie partii przy użyciu klienta Batch Preparation](#)
Klient Batch Preparation umożliwia tworzenie partii szybciej niż jest to możliwe w ramach zwykłego procesu. Gdy użytkownik tworzy nową partię, w polach wyświetlane są wartości z poprzedniej, tak aby w nowej partii można było zachować niektóre pola, a inne w razie potrzeby zmienić.
- [Speed Scan Client: klient szybkiego skanowania](#)
Klient Speed Scan jest używany przez operatorów skanowania do skanowania i przesyłania obrazów.
- [Widok miniatur](#)
Widok miniatur ułatwia nawigację wśród zeskanowanych obrazów. Korzystając z tego widoku, operator skanowania może łatwo i szybko zweryfikować jakość obrazów i w razie potrzeby usunąć niektóre obrazy.
- [Miniatury – przenoszenie i usuwanie](#)
Operator skanowania może zmieniać kolejność miniatur zeskanowanych dokumentów oraz usuwać zeskanowane strony w trakcie procesu skanowania.
- [Odłączanie przeglądarki i przeciągnięcie jej na drugi monitor](#)
Użytkownik może odłączyć przeglądarkę i przeciągnąć ją na drugi monitor.
- [Zachowywanie układu i rozmiarów paneli](#)
Klienty Speed Scan i Speed Index mogą zachowywać układ paneli w konfiguracji dwumonitorowej oraz rozmiary kolumn po zamknięciu i ponownym otwarciu okna przeglądarki.
- [Zmiana kolejności wyboru elementów klawiszem Tab](#)
Użytkownik może określić kolejność wyboru pól klawiszem Tab. Służy do tego projektant panelu.
- [Właściwości skanera](#)
Za pomocą mechanizmu centralnego zarządzania ustawieniami skanowania administrator może zablokować niektóre ustawienia skanera obowiązujące wszystkich użytkowników i zezwolić na przestawianie niektórych ustawień przez operatora skanowania.
- [Dostęp zależny od uprawnień](#)
Funkcje programu Datacap Navigator w zakresie monitorowania zadań z podziałem na role używane są do określania poziomu dostępu na podstawie roli obecnie zalogowanego użytkownika.

- [Definiowanie niestandardowych klawiszy skrótów do przycisków paska narzędzi](#)
Można zdefiniować niestandardowe klawisze skrótów do przycisków paska narzędzi dostępnych w programie Datacap Navigator. Klawisze skrótów są wyświetlane jako podpowiedzi po zatrzymaniu wskaźnika nad przyciskiem w programie Datacap Navigator. Na przykład, gdy użytkownik zatrzyma wskaźnik nad przyciskiem Wyślij, w podpowiedzi wyświetlony zostanie klawisz skrót skojarzony z tym przyciskiem. Jako skróty można też definiować kombinacje klawiszy. Na przykład w przypadku zdefiniowania kombinacji Alt+T jako skrót do przycisku Wyślij naciśnięcie tej kombinacji w programie Datacap Navigator spowoduje wystanie partii do dalszego przetwarzania.
- [Korzystanie z funkcji przesyłania asynchronicznego](#)
Można korzystać z przesyłania asynchronicznego jako opcji domyślnej w kliencie skanowania i kliencie szybkiego skanowania. Nie trzeba czekać z rozpoczęciem następnej czynności na zakończenie przesyłania. Zadania w kolejce przesyłania asynchronicznego są automatycznie wznawiane po błędach, gdy tylko użytkownik z powrotem zaloguje się do aplikacji.
- [Speed Index Client](#)
Klient Speed Index umożliwia użytkownikowi końcowemu (operatorowi indeksowania) szybkie indeksowanie partii oraz dzielenie, łączenie, przenoszenie i usuwanie dokumentów lub stron.
- [Konfiguracja wyświetlania pól indeksu](#)
Użytkownik będący administratorem może modyfikować sposób wyświetlania pól i przycisków na ekranie indeksowania, korzystając z graficznego narzędzia do projektowania.
- [Graficzny projektant formularzy](#)
Możliwe jest stosowanie w formularzach graficznych elementów sterujących, takich jak selektor daty/godziny, pole tekstowe i obszar tekstowy. Ponadto użytkownik za pomocą graficznego projektanta formularzy może modyfikować układ (poprzez przeciąganie kolumn) i rozmiar (w przypadku edytorów).
- [Konfigurowanie pól dokumentu za pomocą panelu dokumentu](#)
Użytkownik może modyfikować pola dokumentu w panelu dokumentu za pomocą projektanta paneli programu Datacap. Za pomocą Panelu dokumentu operator indeksowania może modyfikować właściwości dokumentu w trakcie indeksowania.
- [Dodawanie przycisku do menu paska narzędzi](#)
Istnieje możliwość dodania nowego przycisku do menu paska narzędzi programu Datacap Navigator. Odpowiednie opcje zmienia się w programie IBM Content Navigator.
- [Używanie przycisku wyszukiwania do wypełniania pól dokumentu](#)
Można powiązać przycisk wyszukiwania z regułami i działaniami, aby wprowadzać dane w sposób zautomatyzowany przy użyciu interfejsu usług WWW lub usługi SOAP.
- [Instalowanie wyszukiwania przy użyciu usługi WWW SOAP](#)
Ten temat zawiera przykład wdrożenia wyszukiwania przy użyciu usługi WWW SOAP.
- [Instalowanie wyszukiwania MSSQL](#)
Aby zainstalować przykład wyszukiwania MSSQL, wykonaj następujące czynności.
- [Instalowanie przycisku wyszukiwania w panelu niestandardowym: przykład 1](#)
Aby zainstalować przycisk wyszukiwania w panelu niestandardowym, wykonaj następujące czynności.
- [Instalowanie przycisku wyszukiwania w panelu niestandardowym: przykład 2](#)
Aby zainstalować przycisk wyszukiwania w przykładowym panelu niestandardowym, wykonaj następujące czynności.
- [Usługi danych zewnętrznych](#)
Gdy tworzony jest nowy system, zwykle konieczne jest zintegrowanie ze starszymi, używanymi dotąd systemami. Jednak interfejsy nowego systemu i starych systemów mogą nie być bezpośrednio zgodne ze sobą. W takiej sytuacji należy dodać adapter dla interfejsów.
- [Wdrażanie usług danych zewnętrznych na konsoli serwera Websphere Application Server](#)
Można wdrożyć usługi danych zewnętrznych na konsoli serwera Websphere Application Server. Kompilacja programu Datacap zawiera plik DatacapEDSserver.war.
- [Obsługa uwierzytelniania ADLDS](#)
IBM Datacap umożliwia zarządzanie sesjami użytkowników przy użyciu metody uwierzytelniania opartej na usługach LDS Active Directory (AD LDS) firmy Microsoft.

- [Możliwość pomijania pustych stron](#)

W wersji 9.1.1 i późniejszych programu IBM Datacap możliwe jest pomijanie pustych stron podczas skanowania. Służy do tego opcja Pomiń puste strony w ustawieniach zaawansowanych czynności skanowania. Po skonfigurowaniu tej opcji ustawienie jest zachowywane.

Konfigurowanie programu Datacap Navigator i paneli niestandardowych

Datacap Navigator jest klientem WWW programu opartym na programie IBM® Content Navigator. Klienta Datacap Navigator można skonfigurować, ładując wtyczkę i konfigurując repozytoria oraz pulpity w programie IBM Content Navigator.

Wymagania wstępne

Aby możliwe było korzystanie z klienta Speed Scan i Speed Index, w systemie muszą być zainstalowane programy "IBM Content Navigator 3.0.1" i "Daeja Viewer 5.0.1 IF005".

Konfiguracja programu Datacap Navigator

Aby skonfigurować program Datacap Navigator, należy najpierw zainstalować klienta jako wtyczkę programu IBM Content Navigator. Następnie należy skonfigurować repozytoria odpowiadające aplikacjom programu Datacap. Na koniec można zmodyfikować domyślne pulpity generowane przez program Datacap Navigator.

1. Datacap Navigator instalowany jest jako wtyczka programu IBM Content Navigator. Więcej informacji na temat instalacji programu Datacap Navigator zawiera sekcja [Kroki instalacji programu Datacap Navigator](#).
2. Skonfiguruj repozytoria, aby określić dodatkowe aplikacje Datacap, które można będzie udostępnić użytkownikom klienta Datacap Navigator. Więcej informacji o dodawaniu aplikacji do programu Datacap Navigator lub dodawaniu nowego repozytorium zawiera sekcja [Dodawanie aplikacji do środowiska programu Datacap Navigator](#).
3. Dodaj nowy pulpit. Aby dodać nowy pulpit, wykonaj następujące czynności:
 - a. Zaloguj się do programu Datacap Navigator.
 - b. W lewym panelu kliknij opcję Pulpity > Nowy pulpit.
 - c. Na karcie Ogólne, w polu Nazwa, określ nazwę nowego pulpitu.
 - d. W polu Identyfikator podaj wybrany identyfikator. Ten identyfikator będzie używany w adresie URL do ładowania nowego pulpitu.
 - e. Z listy Repozytorium wybierz repozytorium, które chcesz skonfigurować jako wybierane domyślnie dla tego pulpitu po jego załadowaniu.
Uwaga: Repozytorium z karty Ogólne już jest wymienione na liście Wybrane repozytoria. Można dodać wiele repozytoriów do listy, aby móc przechodzić między aplikacjami, będąc zalogowanym na pulpicie Datacap Navigator.
 - f. Kliknij kartę Układ, wybierz elementy, które mają być wyświetlane na pulpicie, i ułóż je w żądanej kolejności.
 - g. Kliknij opcję Zapisz i zamknij.
4. Skonfiguruj przepływy pracy. Aby skonfigurować przepływy pracy, wykonaj następujące czynności:
 - a. Przejdź do widoku administracyjnego programu Datacap.
 - b. W lewym panelu wybierz pozycję Przepływy pracy.
 - c. Wybierz nazwę przepływu pracy i kliknij przycisk Edytuj.
 - d. Na nowej karcie, na której wyświetlana jest nazwa nowego przepływu pracy, kliknij kartę Zadania .
 - e. Wybierz zadanie programu Navigator, kliknij przycisk Edytuj i wybierz żądane ustawienia konfiguracyjne.
 - f. Kliknij opcję Zapisz i zamknij.

Niestandardowa konfiguracja paneli

Panel jest to ekran służący do wprowadzania danych. Panele są generowane automatycznie przez system i nie wymagają dodatkowej konfiguracji. Można tworzyć własne układy, definiując panele niestandardowe do obsługi czynności weryfikacji, panele rozpoczęcia partii i panele edytorów partii. Można określać układ pól i zmieniać wygląd oraz zachowanie paneli odpowiednio do potrzeb.

Więcej informacji o tworzeniu paneli niestandardowych w programie Datacap Navigator zawiera sekcja [Tworzenie paneli niestandardowych w programie Datacap Navigator](#)

Temat nadrzędny: [Klienty Speed Scan i Speed Index](#)

Klient Batch Preparation

Batch Preparation umożliwia przygotowywanie stron lub partii do szybkiego skanowania.

Użycie klienta Batch Preparation jest pierwszą czynnością w przepływie pracy Zadanie skanowania. Datacap Navigator udostępnia skrót do klienta Batch Preparation, aby można było zrealizować proces jak najszybciej. Za pomocą klienta Batch Preparation Client można na panelu startowym określić różne atrybuty, na przykład identyfikator partii, by utworzyć partię.

Temat nadrzędny: [Klienty Speed Scan i Speed Index](#)

Tworzenie partii przy użyciu klienta Batch Preparation

Klient Batch Preparation umożliwia tworzenie partii szybciej niż jest to możliwe w ramach zwykłego procesu. Gdy użytkownik tworzy nową partię, w polach wyświetlane są wartości z poprzedniej, tak aby w nowej partii można było zachować niektóre pola, a inne w razie potrzeby zmienić.

Procedura

1. Zaloguj się do programu Datacap Navigator.
2. W lewym panelu kliknij pozycję Przygotowanie partii. Zostanie otwarta nowa karta BatchPrep z klientem Batch Preparation.
3. Określ wartości atrybutów, aby utworzyć partię. Podczas tworzenia panelu niestandardowego można używać panelu startowego, aby określić atrybuty, które mają być wyświetlane na stronie przygotowania partii.
Uwaga: Atrybuty systemowe, których zmiana nie jest dozwolona, muszą być podczas tworzenia panelu niestandardowego oznaczone jako przeznaczone tylko do odczytu. Na przykład Id. strony.
4. Aby przetwarzać następną partię, kliknij opcję Zakończ i kontynuuj, a jeśli chcesz wprowadzić bieżącą partię i wyjść z klienta BatchPrep, kliknij opcję Zakończ.

Temat nadrzędny: [Klienty Speed Scan i Speed Index](#)

Speed Scan Client: klient szybkiego skanowania

Klient Speed Scan jest używany przez operatorów skanowania do skanowania i przesyłania obrazów.

Oferuje następujące funkcje i korzyści:

- Możliwość skanowania stron do programu Datacap za pomocą usługi WWW Dynasoft.
- Dwa sposoby wyświetlania obrazów: widok miniatur i całej strony.
- Możliwość centralnego konfigurowania ustawień skanera.

- Obsługa asynchronicznego przesyłania partii: użytkownik może kontynuować pracę na partiach w czasie, gdy trwa przesyłanie obrazów z poprzednich partii.
- Większa wydajność szybkiego skanowania.

Klient Speed Scan Client udostępnia panel Rozpocznij partię służący do wyświetlania i wyboru pól partii. Za pomocą panelu Rozpocznij partię operator skanowania może przesłać obrazy na serwer w trybie nieblokującym i kontynuować skanowanie następnego partii w trakcie przesyłania poprzedniej.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Widok miniatur

Widok miniatur ułatwia nawigację wśród zeskanowanych obrazów. Korzystając z tego widoku, operator skanowania może łatwo i szybko zweryfikować jakość obrazów i w razie potrzeby usunąć niektóre obrazy.

Widok miniatur oferuje następujące korzyści:

- Zajmuje mniej pamięci, a przez to przyczynia się do przyspieszenia skanowania.
- Umożliwia dostrzeżenie pustych obrazów, które mogły zostać przepuszczone mimo zaznaczenia opcji Pomiń puste strony, lub w sytuacji, gdy skaner nie umożliwia pomijania pustych stron.
- [Możliwość przenoszenia, usuwania lub dodawania obrazów](#) z poziomu przeglądarki.

Konfigurowanie widoku miniatur

Aby skonfigurować sposób wyświetlania i położenie miniatur, wykonaj następujące czynności:

1. Przejdź do sekcji Ustawienia.
2. Na karcie Globalne przejdź do sekcji Ustawienia przeglądarki .
3. Z menu listy wybierz opcję Nie wyświetlaj, aby wyłączyć widok miniatur. Jeśli widok miniatur ma być wyświetlany, określ położenie miniatury (np. na górze lub po lewej).

Domyślnie miniatura jest wyrównana do lewej strony.

Elementy sterujące w widoku miniatur

Widok miniatur zawiera następujące elementy sterujące:

- Zmiana powiększenia.
- Za pomocą strzałek w górę i w dół można zmieniać położenie miniatur.
- Opcja usuwania jednym kliknięciem wszystkich stron w widoku miniatur.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Miniatury – przenoszenie i usuwanie

Operator skanowania może zmieniać kolejność miniatur zeskanowanych dokumentów oraz usuwać zeskanowane strony w trakcie procesu skanowania.

W trakcie procesu skanowania można używać menu Działania do zmiany kolejności obrazów zeskanowanych dokumentów poprzez przenoszenie tych obrazów w górę lub w dół.

Jeśli jakość zeskanowanego obrazu jest niezadowalająca, operator skanowania może usunąć ten obraz.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Odłączanie przeglądarki i przeciąganie jej na drugi monitor

Użytkownik może odłączyć przeglądarkę i przeciągnąć ją na drugi monitor.

Uwaga: Można odłączyć tylko przeglądarkę. Nie jest możliwe odłączanie innych paneli. Przeglądarkę można odłączyć od panelu skanowania i indeksowania.

Funkcja ta jest dostępna w następujących trzech obszarach:

- Odłączanie przeglądarki od paneli skanowania i indeksowania
- Ustawienia użytkownika: układ paneli skanowania i indeksowania
- Układ paneli skanowania i indeksowania w trakcie działania programu i zachowywanie tego układu

Odłączanie przeglądarki od paneli skanowania i indeksowania

Gdy użytkownik odłączy przeglądarkę od panelu skanowania lub indeksowania, rozmiar przeglądarki i jej położenie jest zachowywane po zamknięciu i wyjściu z przeglądarki, a następnie ponownym otwarciu partii.

1. **Przypadek A:** użytkownik otwiera przeglądarkę, zamyka ją i ponownie otwiera. Położenie i rozmiar przeglądarki zostają zachowane.
2. **Przypadek B:** użytkownik otwiera przeglądarkę i klika opcję Wstrzymaj lub wprowadza partię, aby wyjść z klienta. Położenie i rozmiar przeglądarki zostają zachowane.

Ustawienia użytkownika

W obszarze Ustawienia użytkownik może zmienić układ i stosować go względem czynności w panelach skanowania i indeksowania.

Uwaga: W panelu indeksowania pole dokumentu jest częścią panelu Rozpocznij/pole.

Trwałość rozmiarów kolumn

Gdy użytkownik zmieni ustawienia panelu skanowania i indeksowania, rozmiar kolumn jest zachowywany po wstrzymaniu, a następnie ponownym otwarciu partii.

Temat nadrzędny: [Klienty Speed Scan i Speed Index](#)

Zachowywanie układu i rozmiarów paneli

Klienty Speed Scan i Speed Index mogą zachowywać układ paneli w konfiguracji dwumonitorowej oraz rozmiary kolumn po zamknięciu i ponownym otwarciu okna przeglądarki.

Gdy w konfiguracji dwumonitorowej użytkownik otworzy przeglądarkę w nowym oknie z klienta Scan Client lub odłączy przeglądarkę i przeciągnie ją na drugi monitor, to przeglądarka będzie wyświetlana w nowym oknie lub na drugim monitorze. Nowe położenie przeglądarki jest zachowywane, gdy użytkownik zamknie i otworzy panel skanowania lub indeksowania.

Podobnie, gdy użytkownik zmieni rozmiary kolumn w kliencie Batch Preparation Client, kliencie skanowania lub kliencie indeksowania, to nowe rozmiary są zachowywane po zamknięciu i ponownym otwarciu klientów.

Ograniczenie:

Gdy w konfiguracji dwumonitorowej użytkownik otworzy partię skanowania lub indeksowania i kliknie opcję Otwórz w nowym oknie na liście Działanie przeglądarki, to przeglądarka Google Chrome wyświetli okno podrzędne na tym samym monitorze, na którym widoczne jest okno nadrzędne, i nie pozwoli na wyświetlenie

okna podrzędnego na drugim monitorze. Jest to ograniczenie przeglądarki wprowadzone ze względów bezpieczeństwa.

W przypadku konfiguracji dwumonitorowej z przeglądarką Microsoft Internet Explorer lub Mozilla Firefox po zamknięciu przeglądarki i ponownym uruchomieniu partii skanowania lub indeksowania położenie na ekranie może nie być dokładnie takie samo, jak w momencie zamykania okna przeglądarki.

W konfiguracji dwumonitorowej z przeglądarką Mozilla Firefox obraz musi być wyświetlany w całości w nowym oknie. W przeciwnym razie jego położenie może zmienić się po kliknięciu przycisku Wstrzymaj. Jest to ograniczenie przeglądarki.

Wskazówka:

W przypadku przeglądarki Mozilla Firefox:

- W konfiguracji dwumonitorowej przeglądarka Mozilla Firefox wyświetla komunikat.
- Kliknij przycisk Edytuj opcje blokowania okien wyskakujących, a następnie kliknij opcję Zezwalaj, aby zezwolić na wyświetlanie na dwóch monitorach.
- Można tak skonfigurować przeglądarkę, aby w trybie dwumonitorowym okno było zamykane automatycznie po kliknięciu przycisku Wstrzymaj. Podstawowy przepływ pracy w kliencie HSSI składa się z następujących kroków:
 1. W przeglądarce Mozilla Firefox wybierz kolejno opcje Opcje > Treść > wyskakujące okna.
 2. Kliknij przycisk Wyjątki.
 3. W polu Adres witryny wprowadź odpowiedni adres URL programu IBM Content Navigator Datacap (na przykład: <http://localhost:9080/navigator>).
 4. Kliknij pozycję Panele.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Zmiana kolejności wyboru elementów klawiszem Tab

Użytkownik może określić kolejność wyboru pól klawiszem Tab. Służy do tego projektant panelu.

Na stronie Ustawienia skanera wyświetlana jest lista konfigurowalnych właściwości skanera, obejmująca ustawienia ogólne, takie jak Automatyczny podajnik dokumentów oraz Tryb duplex. Każdy skaner może mieć własne ustawienia niestandardowe lub działać zgodnie z ogólnymi ustawieniami domyślnymi uzupełnionymi o ewentualne przestonięte ustawienia.

Opcja Kolejność przechodzenia w projektancie paneli służy do modyfikacji kolejności przechodzenia między elementami klawiszem Tab. Funkcja ta dotyczy przede wszystkim pól w panelu Szczegóły pól.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Właściwości skanera

Za pomocą mechanizmu centralnego zarządzania ustawieniami skanowania administrator może zablokować niektóre ustawienia skanera obowiązujące wszystkich użytkowników i zezwolić na przestanianie niektórych ustawień przez operatora skanowania.

Ustawienia czynności wymienione w obszarze "Konfiguracja skanera" są bazowymi właściwościami skanera, które obowiązują, jeśli wszystkie właściwości są zablokowane.

Jeśli niektóre właściwości wymienione w obszarze "Edytowalne możliwości skanera" są odblokowane, użytkownik może zmienić te wartości przed rozpoczęciem skanowania. W takim przypadku zmienione wartości przestają być wartościami właściwości bazowych.

Na karcie "Edytowalne możliwości skanera" wyświetlane są właściwości skanera, które administrator może zablokować lub odblokować. Zaznaczenie pola wyboru przy właściwości oznacza, że użytkownik może zmienić jej wartość. Brak zaznaczenia pola wyboru przy właściwości oznacza, że użytkownik nie może zmienić jej wartości.

Właściwości skanowania i ustawienia blokowania właściwości są zachowywane (trwale).

W sekcji "Edytowalne możliwości skanera" wymienione są następujące właściwości skanera, a obok każdej z nich znajduje się pole wyboru.

- Automatyczny podajnik dokumentów
- Tryb duplex
- Typ pikseli
- Głębina bitowa
- Rozdzielczość (DPI)
- Wielkość papieru
- Format obrazu

Domyślnie wszystkie pola wyboru są niezaznaczone, co oznacza, że właściwości skanera są zablokowane i nie można ich edytować. Jeśli administrator zaznaczy jedno lub więcej pól wyboru, odpowiednie właściwości zostaną odblokowane i operator skanowania będzie mógł je zmienić przed rozpoczęciem skanowania.

Procedura:

1. Zaloguj się jako Administrator, otwórz widok administracyjny programu Datacap i kliknij opcję Przepływy pracy.
2. Wybierz aktywny przepływ pracy.
3. Kliknij kartę **Zadania** i wybierz zadanie, które wymaga zmiany ustawień skanowania.
4. Następnie przejdź do sekcji **Czynności** i wybierz czynność skanowania.
5. Kliknij przycisk **Wyjątki**.
6. Przejdź do sekcji Skanuj. Ta sekcja zawiera sekcje podrzędne "Edytowalne możliwości skanera" i "Widoczność możliwości skanera".
7. Zmieniając zawartość tych sekcji, można uzyskać żądane zachowania skanera, tak jak opisano to w poniższej tabeli.

Tabela 1. Edytowalne możliwości skanera

Pole edytowalnej możliwości skanera zaznaczone?	Widoczność możliwości skanera	Efekt	Opis
Tak	Tak	Widoczne i odblokowane	Ustawienie jest wyświetlane i może być edytowane
Tak	Nie	Widoczne i odblokowane	Ustawienie jest wyświetlane i może być edytowane
Nie	Tak	Widoczne i zablokowane	Ustawienie jest wyświetlane, ale nie może być edytowane
Nie	Nie	Niewidoczne	Ustawienie nie jest wyświetlane i nie może być edytowane

Ważne: Jeśli właściwość skanera jest edytowalna, to zmiany wprowadzone przez użytkownika przestają bazować na właściwościach dla czynności/zadania.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Dostęp zależny od uprawnień

Funkcje programu Datacap Navigator w zakresie monitorowania zadań z podziałem na role używane są do określania poziomu dostępu na podstawie roli obecnie zalogowanego użytkownika.

Tylko administrator ma dostęp do opcji Przepływ pracy. Pozostali użytkownicy (na przykład operatorzy skanowania) nie powinni mieć dostępu do opcji Przepływ pracy. Gdy operator skanowania zaloguje się do pulpitu programu Datacap Navigator, opcja Przepływ pracy będzie niedostępna, zatem zwykły użytkownik nie może skonfigurować ustawień czynności.

Ważne: Należy dopilnować, aby tylko administratorzy Datacap mieli dostęp do opcji Przepływ pracy.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Definiowanie niestandardowych klawiszy skrótów do przycisków paska narzędzi

Można zdefiniować niestandardowe klawisze skrótów do przycisków paska narzędzi dostępnych w programie Datacap Navigator. Klawisze skrótów są wyświetlane jako podpowiedzi po zatrzymaniu wskaźnika nad przyciskiem w programie Datacap Navigator. Na przykład, gdy użytkownik zatrzyma wskaźnik nad przyciskiem Wyślij, w podpowiedzi wyświetlony zostanie klawisz skrótów skojarzony z tym przyciskiem. Jako skrótów można też definiować kombinacje klawiszy. Na przykład w przypadku zdefiniowania kombinacji Alt+T jako skrótów do przycisku Wyślij naciśnięcie tej kombinacji w programie Datacap Navigator spowoduje wystanie partii do dalszego przetwarzania.

Procedura

1. Zaloguj się do pulpitu administracyjnego programu IBM Content Navigator.
2. W lewym panelu kliknij opcję Wtyczki.
3. Na liście Nazwy kliknij dwukrotnie pozycję Datacap Navigator.
4. Na stronie Datacap Navigator kliknij opcję Dostosuj klawisze skrótów. Zostanie otwarte okno Niestandardowe klawisze skrótów.
5. W oknie Niestandardowe klawisze skrótów domyślnie wybrana jest karta Najczęściej wykonywane. Określ klawisze skrótów do przycisków, takich jak Wyślij, Anuluj itd.
6. Jeśli chcesz utworzyć klawisze skrótów do przycisków w następujących procesach, klikaj odpowiednie karty i definiuj klawisze skrótów:

Uwaga: Aby zdefiniować klawisz skrótów, naciśnij klawisz Alt i klawisz, który ma tworzyć z nim kombinację. W polu wyświetlana jest kombinacja Alt+T.

- o Batch Prep
- o Speed Scan
- o Speed Index
- o Klasyfikowanie
- o Weryfikacja
- o Struktura partii
- o Przeglądarka

Uwaga: Na karcie Klawisze skrótów przeglądarki nie można określić klawiszy skrótów następujących przycisków. Klawisze skrótów są już przypisane domyślnie:

- Powiększ

- Pomniejsz
- Dopasuj do szerokości
- Dopasuj do wysokości
- Poprzednia strona
- Następna strona

Uwaga: Można też definiować kombinacje klawiszy jako skróty.

7. Po zdefiniowaniu wszystkich klawiszy skrótów kliknij przycisk Zapisz.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Korzystanie z funkcji przesyłania asynchronicznego

Można korzystać z przesyłania asynchronicznego jako opcji domyślnej w kliencie skanowania i kliencie szybkiego skanowania. Nie trzeba czekać z rozpoczęciem następnej czynności na zakończenie przesyłania. Zadania w kolejce przesyłania asynchronicznego są automatycznie wznawiane po błędach, gdy tylko użytkownik z powrotem zaloguje się do aplikacji.

Procedura

1. Zaloguj się do programu Datacap Navigator.
2. Przejdź do widoku administracyjnego programu Datacap i kliknij pozycję Przepływy pracy
3. Kliknij dwukrotnie przepływ pracy na liście i kliknij kartę **Zadania**.
4. Kliknij dwukrotnie dowolne zadanie, a następnie kartę Czynności.
5. Kliknij dwukrotnie dowolną czynność, przejdź do panelu Rozpocznij partię i zaznacz pole wyboru Przesyłanie asynchroniczne.

Jeśli włączona jest funkcja Przesyłanie asynchroniczne, proces będzie z niej korzystał.

Jeśli użytkownik nie wybierze opcji Przesyłanie asynchroniczne ani Prześlij natychmiast, to zadanie przesyłania trzeba będzie dodać ręcznie.

Uwaga: Zaznacz pole wyboru Prześlij natychmiast, aby nakazać rozpoczęcie czynności przesyłania od razu po pomyślnym zakończeniu czynności skanowania dla partii.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Speed Index Client

Klient Speed Index umożliwia użytkownikowi końcowemu (operatorowi indeksowania) szybkie indeksowanie partii oraz dzielenie, łączenie, przenoszenie i usuwanie dokumentów lub stron.

Klient Speed Index umożliwia użytkownikowi (operatorowi indeksowania) wykonywanie następujących działań:

- Wybór partii do indeksowania przy użyciu zapytania.
Uwaga: Klient Speed Index obsługuje tylko wprowadzanie danych w polach na poziomie dokumentu, a nie na poziomie strony.
- Przetwarzanie dokumentów z zachowaniem priorytetów zdefiniowanych przez administratora.
- Szybkie przetwarzanie dokumentów za pomocą klawiatury.
- Kierowanie problemów na poziomie partii do innych kolejek celem dokonania przeglądu i ponownego skanowania
- Weryfikowanie jakości zeskanowanych obrazów i stosowanie podstawowych korekt obrazów w celu poprawy ich czytelności.

Klient Speed Index umożliwia także administratorom wydziałowym monitorowanie produktywności pracy operatorów indeksowania.

Szybki proces indeksowania:

1. Operator indeksowania wybiera czynność indeksowania.
2. Wyświetlany jest formularz wyboru partii z kryterium wyboru zapytania.
3. Operator indeksowania wprowadza kryteria wyszukiwania partii.
4. Może wystąpić jedna z dwóch sytuacji:
 - a. Operator klika przycisk Uruchom: wybierana jest partia o najwyższym priorytecie spełniająca kryteria oraz dostępna dla grupy uprawnień. Należy przejść do kroku 5.
 - b. Operator klika przycisk Lista: zostaje wyświetlona lista partii spełniająca kryteria oraz dostępnych dla grupy uprawnień, posortowana wg priorytetu. Operator wybiera partię z listy. Należy przejść do kroku 5.
5. Zostaje wyświetlony interfejs użytkownika indeksowania z widoczną strukturą partii, panelem indeksowania i pierwszym obrazem.

Temat nadrzędny: [Klienty Speed Scan i Speed Index](#)

Konfiguracja wyświetlania pól indeksu

Użytkownik będący administratorem może modyfikować sposób wyświetlania pól i przycisków na ekranie indeksowania, korzystając z graficznego narzędzia do projektowania.

Administrator może skonfigurować kolejność pól i przycisków wyświetlanych na stronach Batch Preparation, Speed Scan i Speed Index, aby ułatwić wszystkim użytkownikom korzystanie z aplikacji.

Cechy wyróżniające:

- Użytkownik może [modyfikować pola dokumentu w panelu dokumentu za pomocą projektanta paneli programu Datacap](#). Pola dokumentu są wyświetlane na stronie indeksu.
- Użytkownik może skonfigurować przycisk wyszukiwania w projektancie paneli programu Datacap. Monit może być teraz przyciskiem, a nie odsyłaczem hipertekstowym.
- Przycisk wyszukiwania jest obsługiwany tylko w polach Pole tekstowe i Obszar tekstowy. Inne typy pól nie obsługują przycisku wyszukiwania.
- Użytkownik może [dodać przycisk do menu paska narzędzi](#). Odpowiednie modyfikacje wprowadza się z pulpitu administracyjnego IBM Content Navigator.
- **Gdy użytkownik kliknie dokument w drzewie partii:** w panelu Szczegóły pól wyświetlane są pola na poziomie dokumentu. W przeglądarce wyświetlany jest obraz pierwszej strony tego dokumentu. Jeśli dokument nie zawiera żadnych stron, przeglądarka przedstawia pustą stronę lub symbol zastępczy oznaczający brak obrazu.
- **Gdy użytkownik kliknie stronę w drzewie partii:** w panelu Szczegóły pól wyświetlane są pola dokumentu, do którego należy strona. W przeglądarce wyświetlany jest obraz strony.

Temat nadrzędny: [Klienty Speed Scan i Speed Index](#)

Graficzny projektant formularzy

Możliwe jest stosowanie w formularzach graficznych elementów sterujących, takich jak selektor daty/godziny, pole tekstowe i obszar tekstowy. Ponadto użytkownik za pomocą graficznego projektanta formularzy może modyfikować układ (poprzez przeciąganie kolumn) i rozmiar (w przypadku edytorów).

Dostępne są następujące nowe funkcje:

- **Obsługa przycisku wyszukiwania:** wyłącznie w polach tekstowych i obszarach tekstowych.
- **Zmiana kolejności wyboru elementów klawiszem Tab**
- Nowy typ panelu: Panel dokumentu. W panelu indeksowania wyświetlany jest Panel startowy i Panel dokumentu.

Procedura:

1. Zaloguj się do programu Datacap Navigator jako użytkownik będący administratorem.
2. Otwórz widok administracyjny programu Datacap.
3. Wśród opcji na lewym panelu kliknij pozycję Panele.
4. Kliknij przycisk Nowy panel. Zostanie otwarty projektant paneli.
5. W sekcji Ustawienia można określić klawisz skrótów działania wyszukiwania.

Uwaga:

- Definiując klawisz skrótów dla przycisku wyszukiwania, należy zwrócić uwagę, by nie kolidował on z klawiszem skrótów paska menu głównego (który określa się za pomocą menu Konfiguruj klawisze skrótów) i klawiszami skrótów przeglądarki.
- Klawisz skrótów zdefiniowany w tym miejscu i na pasku narzędzi programu IBM Content Navigator ma wyższy priorytet niż klawisze skrótów przeglądarki.

Tabela 1. Klawisze skrótów przeglądarki

Działanie	Klawisz skrótów
Pokaż miniatury	F3
Pokaż dwie strony	F4
Pokaż miniatury: z lewej	F5
Pokaż miniatury: na dole	F6
Pokaż miniatury: z prawej	F7
Pokaż miniatury: na górze	F8

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Konfigurowanie pól dokumentu za pomocą panelu dokumentu

Użytkownik może modyfikować pola dokumentu w panelu dokumentu za pomocą projektanta paneli programu Datacap. Za pomocą Panelu dokumentu operator indeksowania może modyfikować właściwości dokumentu w trakcie indeksowania.

- Datacap Studio udostępnia typ dokumentu o nazwie Document. Ten typ dokumentu zawiera dwa typy stron: Strona i Strona końcowa.
- Typ dokumentu zawiera także Pola dokumentu, takie jak Last, Middle, First i tak dalej. Są to domyślne pola dokumentu, które można wybrać w panelu indeksowania.

W niniejszej procedurze opisano sposób tworzenia niestandardowego panelu dokumentów i wyświetlania niestandardowych pól dokumentu w panelu indeksowania – w miejsce domyślnych pól utworzonych w Datacap Studio.

1. Otwórz widok administracyjny programu Datacap.
2. Kliknij pozycję Panele.
3. Kliknij przycisk Nowy panel i z listy wybierz pozycję Panel dokumentu.
4. Jako Typ strony wybierz „Dokument” i podaj nazwę w polu Nazwa.
5. Wprowadź wymagane modyfikacje, aby możliwe było odróżnienie domyślnego pola dokumentu od pól niestandardowych.

Uwaga: Użytkownik [może skonfigurować przycisk wyszukiwania w projektancie paneli programu Datacap](#). W takim przypadku w monicie zamiast odsyłacza hipertekstowego będzie wyświetlany przycisk.

6. Zapisz zmiany i zamknij.
7. Przejdź do obszaru Przepływy pracy i otwórz kartę Zaawansowane w sekcji Zadania > Skanowanie > Czynności > Indeksowanie.
8. Na karcie Zaawansowane wprowadź nowy, utworzony wcześniej panel niestandardowy, zapisz zmiany i zamknij okno.
9. Wyloguj się i zaloguj ponownie. Przejdź do czynności indeksowania. Po podzieleniu dokumentu i przypisaniu mu typu można przypisać obrazy do typu strony powiązanego z wybranym typem dokumentu
10. Można przetaczać się między dokumentami widocznymi na stronie. Gdy wybrana jest strona podrzędna dokumentu, zachowuje ona wartość z dokumentu nadrzędnego.
 - Gdy użytkownik kliknie dokument w drzewie partii w panelu Szczegóły pól wyświetlane są pola na poziomie dokumentu, a w przeglądarce widoczny jest obraz pierwszej strony tego dokumentu.
 - Jeśli dokument nie zawiera żadnych stron, przeglądarka przedstawia pustą stronę lub symbol zastępczy oznaczający brak obrazu.
 - Gdy użytkownik kliknie stronę w drzewie partii, w panelu Szczegóły pól wyświetlane są pola dokumentu, do którego należy strona, a w przeglądarce widoczny jest obraz strony.

Uwaga: Przycisk wyszukiwania obsługuje tylko pola tekstowe i obszary tekstowe. Inne typy pól nie są obsługiwane przez przycisk wyszukiwania.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Dodawanie przycisku do menu paska narzędzi

Istnieje możliwość dodania nowego przycisku do menu paska narzędzi programu Datacap Navigator. Odpowiednie opcje zmienia się w programie IBM Content Navigator.

Odpowiednie modyfikacje wprowadza się z pulpitu administracyjnego IBM Content Navigator.

1. Otwórz konsolę administracyjną programu IBM Content Navigator.
2. W lewym panelu kliknij opcję Menu. W obszarze filtrowania wprowadź `datacap` jako filtr wyszukiwania.
Uwaga:

Domyślnych menu nie można edytować, ale można utworzyć własne menu, korzystając z działania Kopiuj.
3. Edytuj pasek narzędzi i dodaj funkcję, która ma się znaleźć w nowym menu. Na przykład dodaj funkcję Zmień ustawienia użytkownika.
4. Zapisz zmiany i zamknij.
5. W lewym panelu konsoli administracyjnej IBM Content Navigator kliknij opcję Pulpity.
6. Wybierz pulpit, który chcesz edytować, a następnie kliknij kartę Menu.
7. Wybierz utworzone samodzielnie menu paska narzędzi i użyj go w kliencie Batch Prep lub innym panelu.
8. Zapisz zmiany i zamknij.
9. Ponownie zaloguj się do aplikacji Datacap, aby zweryfikować zmiany.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Używanie przycisku wyszukiwania do wypełniania pól dokumentu

Można powiązać przycisk wyszukiwania z regułami i działaniami, aby wprowadzać dane w sposób zautomatyzowany przy użyciu interfejsu usług WWW lub usługi SOAP.

Aby dodać i skonfigurować przycisk wyszukiwania za pomocą panelu dokumentu, wykonaj następujące czynności:

1. Otwórz widok administracyjny programu Datacap.
2. Kliknij pozycję Panele.
3. Kliknij przycisk Nowy panel. Z listy wybierz pozycję Panel dokumentu.
4. Wybierz odpowiedni Typ strony z listy.
5. W polu Nazwa wprowadź nazwę nowego panelu.
6. W sekcji Ustawienia kliknij opcję Styl odsyłacza wyszukiwania i z listy wybierz opcję Przycisk.

Informacje na temat definiowania przycisku wyszukiwania zawierają następujące tematy:

- [Instalowanie przycisku wyszukiwania w panelu niestandardowym: przykład 1](#)
- [Instalowanie przycisku wyszukiwania w panelu niestandardowym: przykład 2](#)
- [Instalowanie wyszukiwania przy użyciu usługi WWW SOAP.](#)
- [Instalowanie wyszukiwania MSSQL.](#)
- Skonfiguruj (za pośrednictwem projektanta), sprawdź i przetestuj wyszukiwanie za pomocą usługi WWW REST.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Instalowanie wyszukiwania przy użyciu usługi WWW SOAP

Ten temat zawiera przykład wdrożenia wyszukiwania przy użyciu usługi WWW SOAP.

O tym zadaniu

Uwaga: Następujący odsyłacz prowadzi do serwisu służącego do testowania usługi WWW SOAP: "Airport Information Webservice" <http://www.websvc.net/airport.asmx>

Procedura

1. Zaloguj się do programu Datacap Studio.
 - a. Utwórz nowy profil czynności wywołujący działania usługi WWW. Należy użyć następujących wartości:

WsSoapSetEnvelope parameter value:

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:xsi
="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://
schemas.xmlsoap.org/soap/envelope/"><soap:Body>
<getAirportInformationByAirportCode
xmlns="http://www.webserviceX.NET"><airportCode>SNA</airportCode>
</getAirportInformationByAirportCode>
</soap:Body></soap:Envelope>
```

WsSoapGetValues parameter value:

```
http://www.websvc.net/airport.asmx, ' ', http://www.webserviceX.
NET, getAirportInformationByAirportCodeResponse, ' ', ' ', c:\\out2.xml
```

- i. Utwórz nowy zestaw reguł Airport_SOAP używany w profilu czynności "AirportData".
- ii. Utwórz nowy profil czynności SoapAirportData.

- iii. Dodaj regułę do poziomu partii DCO, korzystając z polecenia *Nazwa przepływu pracy > Otwórz*.
 - b. Utwórz trzy pola na poziomie partii:
 - B_SOAP_AirportCode
 - B_SOAP_CityOrAirportName
 - B_SOAP_RunwayLengthFeet
2. Zaloguj się do programu Datacap Navigator.
 3. Kliknij pozycję Panele. Utwórz "Panel startowy", na przykład "lookup test". Należy użyć następujących wartości.
 - o **Styl odsyłacza wyszukiwania:** Button
 - o **Reguła:** SoapAirportData
 - o **Dane wejściowe do mapowania:**

```
B_SOAP_CityOrAirportName:NewDataSet.Table[0].CityOrAirportName,B_SOAP_RunwayLengthFeet:NewDataSet.Table[0].RunwayLengthFeet,B_SOAP_AirportCode:NewDataSet.Table[0].AirportCode
```
 4. Uruchom klienta skanowania i kliknij pozycję B_SOAP_AirportCode. Zostanie wyświetlone okno.
 5. Kliknij dwukrotnie jeden z wyników. Pola zostaną zaktualizowane.

Info (getAirportInformationByAirportCodeResult)

JSON:

```
{ "NewDataSet": { "Table": [ { "LongitudeEperW": "W",
"LongitudeMinute": "52", "LongitudeDegree": "117",
"CountryAbbrviation": "US", "linebreakLatitudeSecond": "0", "LatitudeMinute": "40",
"LatitudeDegree": "33", "LatitudeNpeers": "N", "CityOrAirportName":
"SANTA ANA WAYNE INTL", "RunwayLengthFeet": "5700", "RunwayElevationFeet": "54",
"Country": "United States", "CountryCode": "91", "AirportCode":
"SNA", "LongitudeSeconds": "0", "GMTOffset": "8" },
{ "LongitudeEperW": "W", "LongitudeMinute":
"52", "LongitudeDegree": "117", "CountryAbbrviation": "US", "LatitudeSecond":
"0", "LatitudeMinute": "40", "LatitudeDegree": "33", "LatitudeNpeers":
"N", "CityOrAirportName": "SANTA ANA WAYNE INTL", "RunwayLengthFeet":
"5700", "RunwayElevationFeet": "54", "Country": "United States", "CountryCode": "91",
"AirportCode": "SNA", "LongitudeSeconds": "0", "GMTOffset": "8" } ] } }
```

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Instalowanie wyszukiwania MSSQL

Aby zainstalować przykład wyszukiwania MSSQL, wykonaj następujące czynności.

Procedura

1. Zaloguj się do programu Datacap Studio.
 - a. Utwórz nowy profil czynności wywołujący działania usługi WWW.
 - i. Utwórz nowy zestaw reguł Airport_SQL używany w profilu czynności "DBLookupData".

Wartość parametru OpenConneciton:

```
Provider=SQLOLEDB;Server=9.30.94.208;
Database=Airport;UID=sa;PWD=Passw0rd;
```

rrSet parameter value:

```
"Name,Code,Weather,State,Terminal",@B.flist"
```

Wartość parametru ExecuteSQLEx:

```
"select * from Airport_Data WHERE Weather  
= 'SUNNY';", True, @B.TransactionDatabaseLookup
```

- ii. Utwórz nowy profil czynności "DBLookupData".
 - iii. Dodaj regułę do poziomu partii DCO, korzystając z polecenia *Nazwa przepływu pracy* > Otwórz.
- b. Utwórz 3 pola na poziomie partii: B_SQL_Airport_City, B_SQL_Airport_Acr i B_SQL_Airport_Runway.
2. Zaloguj się do programu Datacap Navigator i kliknij pozycję Panele.
 3. Utwórz "Panel startowy", na przykład "lookuptest".

Styl odsyłacza wyszukiwania: Button

Reguła: DBLookupData

Dane wejściowe do mapowania:

```
B_SQL_Airport_City:Results.Name,B_SQL_Airport_Acr:  
Results.Code,B_SQL_Airport_Runway:Results.Terminal
```

4. Uruchom klienta skanowania i kliknij pozycję B_SQL_Airport_City. Zostanie wyświetlone okno.
5. Kliknij dwukrotnie jeden z wyników. Pola zostaną zaktualizowane.
Info (JSON data).

```
{"Results":{"State":"NC","Weather":"SUNNY","Name":  
"CHARLOTTE","Terminal":"22","Code":"CLT"}}
```

Uwagi: Nazwy kolumn służą do odwzorowywania (mapowania) na wyniki zwrócone przez instrukcję select w działaniu ExecuteSQLEx. W tym przykładzie instrukcja select zwraca:

```
"CHARLOTTE CLT SUNNY NC 22"
```

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Instalowanie przycisku wyszukiwania w panelu niestandardowym: przykład 1

Aby zainstalować przycisk wyszukiwania w panelu niestandardowym, wykonaj następujące czynności.

Procedura

1. Zaloguj się do programu Datacap Studio.
 - a. Utwórz nowy profil czynności wywołujący działania usługi WWW.
 - i. Utwórz nowy zestaw reguł używany w profilu czynności "AirportData".
 - ii. Utwórz nowy profil czynności "AirportData".
 - iii. Dodaj regułę do poziomu partii, korzystając z polecenia *Nazwa przepływu pracy* > Otwórz.
 - b. Utwórz dwa pola na poziomie partii: B_Airport_State i B_Airport_Name.
2. Zaloguj się do programu Datacap Navigator i kliknij pozycję Panele.
3. Utwórz "Panel startowy", na przykład "lookupbuttonp". Dane wejściowe do mapowania:
B_Airport_State:AirportStatus.State i B_Airport_Name:AirportStatus.Name .
4. Zastosuj panel startowy w kliencie skanowania.
5. Uruchom klienta skanowania i kliknij pozycję B_Airport_Name. Zostanie wyświetlone okno.
6. Kliknij dwukrotnie jeden z wyników.
Info: (dane o porcie lotniczym San Francisco)

```
<?xml version="1.0" encoding="UTF-8"?><AirportStatus><Delay>>false</Delay>
<IATA>SFO</IATA><State>California</State><Name>San Francisco
International</Name>
<Weather><Visibility>10.00</Visibility><Weather>Niewielkie
zachmurzenie</Weather>
<Meta><Credit>NOAA's National Weather Ser-vice</Credit>
<Updated>7:56 PM Lo-cal</Updated><Url>http://weather.gov/</Url>
</Meta><Temp>63.0 F (17.2 C)</Temp><Wind>Zachodni 29,61 km/h</Wind>
</Weather><ICAO>KSFO</ICAO><City>San Francisco</City><Status>
<Reason>Brak informacji o opóźnieniach na tym lotnisku.</Reason><ClosureBegin>
</ClosureBegin>
<EndTime></EndTime><MinDelay></MinDelay><AvgDelay></AvgDelay><MaxDelay>
</MaxDelay>
<ClosureEnd></ClosureEnd><Trend></Trend><Type></Type></Status></AirportStatus>
```

Temat nadrzędny: Klienty Speed Scan i Speed Index

Instalowanie przycisku wyszukiwania w panelu niestandardowym: przykład 2

Aby zainstalować przycisk wyszukiwania w przykładowym panelu niestandardowym, wykonaj następujące czynności.

Procedura

1. Zaloguj się do programu Datacap Studio.
 - a. Utwórz nowy profil czynności wywołujący działania usługi WWW.
 - i. Utwórz nowy zestaw reguł używany w profilu czynności "AirportData".
 - ii. Utwórz nowy profil czynności "AirportData."
 - iii. Dodaj regułę do poziomu partii, korzystając z polecenia *Nazwa przepływu pracy* > Otwórz.
 - b. Utwórz dwa pola na poziomie partii: B_Airport_State i B_Airport_Name.
2. Zaloguj się do programu Datacap Navigator i kliknij pozycję Panele.
3. Utwórz "Panel startowy", na przykład "lookupbuttonp". Dane wejściowe do mapowania: B_Airport_State:AirportStatus.State i B_Airport_Name:AirportStatus.Name .
4. Zastosuj panel startowy w kliencie skanowania.
5. Uruchom klienta skanowania i kliknij pozycję B_Airport-Name. Zostanie wyświetlone okno.
6. Kliknij dwukrotnie jeden z wyników. Pola zostaną zaktualizowane.

Info: (dane o porcie lotniczym San Francisco)

```
<?xml version="1.0" encoding="UTF-8"?><AirportStatus><Delay>>false</Delay>
<IATA>SFO</IATA><State>California</State><Name>San Francisco
International</Name>
<Weather><Visibility>10.00</Visibility><Weather>Niewielkie
zachmurzenie</Weather>
<Meta><Credit>NOAA's National Weather Ser-vice</Credit>
<Updated>7:56 PM Lo-cal</Updated><Url>http://weather.gov/</Url>
</Meta><Temp>63.0 F (17.2 C)</Temp><Wind>Zachodni 29,61 km/h</Wind>
</Weather><ICAO>KSFO</ICAO><City>San Francisco</City>
<Status><Reason>Brak informacji o opóźnieniach na tym lotnisku.</Reason>
<ClosureBegin></ClosureBegin><EndTime></EndTime><MinDelay></MinDelay>
<AvgDelay></AvgDelay><MaxDelay></MaxDelay><ClosureEnd></ClosureEnd>
<Trend></Trend><Type></Type></Status></AirportStatus>
```

Temat nadrzędny: Klienty Speed Scan i Speed Index

Usługi danych zewnętrznych

Gdy tworzony jest nowy system, zwykle konieczne jest zintegrowanie ze starszymi, używanymi dotąd systemami. Jednak interfejsy nowego systemu i starych systemów mogą nie być bezpośrednio zgodne ze sobą. W takiej sytuacji należy dodać adapter dla interfejsów.

Dostępne są następujące strategie dodawania adaptera:

Dodawanie po stronie klienta

Jest to proste rozwiązanie, ale wiązą się z nim następujące czynniki ryzyka:

1. **Ryzyko nieskuteczności zabezpieczeń:** ponieważ klient komunikuje się bezpośrednio ze starym systemem, komunikacja wychodzi poza firewall i może zostać przechwycona po stronie przeglądarki.
2. **Ryzyko uwierzytelniania:** to ryzyko może nie występować na kliencie działającym w konfiguracji klient/serwer. Jednak w przypadku klienta w konfiguracji przeglądarka/serwer nie jest możliwe skompilowanie kodu JavaScript. Dlatego zapisywanie danych uwierzytelniających po stronie klienta nie jest bezpieczne.
3. **Ograniczenie komunikacji międzydomenowej narzucone przez przeglądarkę:** ze względów bezpieczeństwa przeglądarka nie zezwala na pobieranie danych z innej domeny, co utrudnia klientowi bezpośrednio uzyskiwanie danych ze starego systemu.
4. **Wymóg odseparowania:** zmiana w starym systemie prowadzi do zmiany we wszystkich klientach i konieczności powtórzenia testów.

Dodawanie po stronie serwera

Ta struktura działa prawidłowo, ale wymusza zbyt ścisłe powiązanie komponentów. Oznacza to, że zmiany w interfejsie startego systemu wymuszają wprowadzenie modyfikacji po stronie serwera.

Dodawanie w odrębnej aplikacji

Nowa aplikacja pobiera dane ze starego systemu i pełni rolę opakowania wymaganego przez interfejs nowego systemu. Jest to dobre rozwiązanie, gdy chcemy oddzielić stary system od nowego. Zmiana starego systemu wymusza jedynie zmianę odrębnej aplikacji, ale nie wymusza zmiany nowego systemu. Usługi danych zewnętrznych są realizowane w całości przez odrębną aplikację.

Ponieważ niestandardowy panel utworzony przez PVD jest wyłącznie kodem HTML uruchamianym po stronie przeglądarki, nie jest bezpieczne zapisywanie adresów URL usług WWW, odsyłaczy do bazy danych lub danych uwierzytelniających.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Wdrażanie usług danych zewnętrznych na konsoli serwera Websphere Application Server

Można wdrożyć usługi danych zewnętrznych na konsoli serwera Websphere Application Server. Kompilacja programu Datacap zawiera plik DatacapEDSserver.war.

Procedura

1. Otwórz WebSphere Application Server.
2. Kliknij przycisk Nowa aplikacja.

- Wybierz opcję Local file system (Lokalny system plików) i przejdź do pliku DatacapEDSserver.war, a następnie kliknij przycisk Next (Dalej).
- Wybierz opcję Fast Path - Prompt only when additional information is required (Szybka ścieżka - monituj tylko wtedy, gdy są potrzebne dodatkowe informacje) i kliknij przycisk Next (Dalej).
- W polu Context Root (Kontekstowy katalog główny) wpisz /DatacapEDSService, kliknij przycisk Next (Dalej) i dokończ instalację aplikacji.
- Zaloguj się do programu Datacap Navigator.
- Utwórz wtyczkę.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Obsługa uwierzytelniania ADLDS

IBM Datacap umożliwia zarządzanie sesjami użytkowników przy użyciu metody uwierzytelniania opartej na usługach LDS Active Directory (AD LDS) firmy Microsoft.

Mechanizm ten sprawia, że użytkownicy nie uzyskują dostępu do zadań innych użytkowników, nawet jeśli spróbują zmodyfikować identyfikator QueueID w adresie URL na taki, który przypisano innemu użytkownikowi.

Uwaga: Implementacja uwierzytelniania AD LDS w programie IBM Datacap wymaga zastosowania protokołu Level Lightweight Directory Access Protocol (LLLDAP). Wtyczki Datacap ADLDS należy używać tylko z katalogiem globalnym. W przypadku używania ISAM zaleca się stosowanie protokołu LLLDAP.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Możliwość pomijania pustych stron

W wersji 9.1.1 i późniejszych programu IBM Datacap możliwe jest pomijanie pustych stron podczas skanowania. Służy do tego opcja Pomiń puste strony w ustawieniach zaawansowanych czynności skanowania. Po skonfigurowaniu tej opcji ustawienie jest zachowywane.

Nie wszystkie skanery obsługują opcję pomijania pustych stron. W przypadku skanerów obsługujących tę funkcję opcja dostępna jest we właściwościach skanera, na przykład właściwości Strona skanowania, która jako opcje do wyboru zawiera np. DupleksPomiń puste strony.

Po wybraniu opcji Dupleks skaner realizuje skanowanie dwustronne. Jeśli wybrana zostanie opcja Pomiń puste strony, skaner będzie skanował dwustronnie w taki sposób, że jeśli jedna ze stron będzie pusta, to nie zostanie uwzględniona w wynikach skanowania.

W poniższej tabeli przedstawiono oczekiwane rezultaty różnych kombinacji ustawień opcji Dupleks, Pomiń puste strony oraz możliwości skanera.

Tabela 1. Ustawienia skanera

Opcja Dupleks	Opcja Pomiń puste strony	Oczekiwany rezultat
Wybrana	Niewybrana	Wyniki skanowania są dwustronne.
Wybrana	Wybrana	Jeśli skaner nie obsługuje pomijania pustych stron, wyniki skanowania są dwustronne. Jeśli skaner obsługuje pomijanie pustych stron, puste strony nie są uwzględniane w wyniku skanowania.

Opcja Dupleks	Opcja Pomiń puste strony	Oczekiwany rezultat
Niewybrana	Niewybrana	Wyniki skanowania są jednostronne.
Niewybrana	Wybrana	Jeśli skaner nie obsługuje pomijania pustych stron, wyniki skanowania są jednostronne. Jeśli skaner nie obsługuje pomijania pustych stron, wyniki skanowania są dwustronne.

Właściwości skanera klienta Speed Scan odzwierciedlają domyślne ustawienia czynności lub niestandardowe ustawienia skanera, jeśli wybrane źródło skanowania takie ustawienia oferuje.

Gdy użytkownik wybierze skaner mający ustawienia niestandardowe, właściwości w interfejsie użytkownika klienta skanowania wypełniane są na podstawie domyślnych ustawień skanera. Jeśli wybrany skaner nie ma ustawień niestandardowych, obowiązują domyślne ogólne ustawienia czynności skanowania.

Użytkownik należący grupy mającej uprawnienia do edycji może przed rozpoczęciem skanowania zmodyfikować właściwości za pomocą interfejsu użytkownika. Zmiany dokonane w interfejsie użytkownika nie zmieniają ustawień domyślnych, ale przestają być podczas skanowania.

Wartości wyświetlane w interfejsie użytkownika klienta Speed Scan są tymi samymi, które obowiązują przy skanowaniu, niezależnie od tego, który operator je wykonuje.

Temat nadrzędny: [Klienci Speed Scan i Speed Index](#)

Image scans

You can scan your source images and collect them into batches by using Datacap Navigator. To open the scan view, click the scan shortcut in the Datacap Navigator client.

The Source field displays the directory from which the source files are scanned. You can change the source directory by clicking Browse and choosing a new source directory.

If your application includes a Start Batch panel, click Start Batch Panel to enter any data specific to the batch that you want to collect.

You can use Hold or Cancel to stop processing the batch. While holding the batch leaves it available to the same user for later processing, canceling will also delete both the batch and all of its data from the system.

When you are finished scanning, submit the batch to complete the task.

Restriction: If Datacap Navigator is configured to upload images automatically after submitting a scan, the `Multiple` option is not supported for the Program setting of the upload task.

Parent topic: [Administering Datacap Navigator](#)

Batch verification

You run the Verify task in Datacap Navigator to ensure that your data was accurately captured and recognized by your application. In a production environment, Datacap verification tasks are run manually to identify and correct potential data problems before you upload the images to Datacap.

The Batch Structure pane shows the batch hierarchy. You can use the controls to manually restructure, reorder, and check the integrity of the batch.

To see the details of a specific page, you can click an image to load that page's fields in the Field Detail pane. When a field fails validation, it has a red background, and you must correct the data before you can complete the batch. If the field has an associated error message, select the field to display the message.

When the recognition results for a field show a low confidence with a yellow background, you can click to update the contents. Alternatively, you can click outside of the field to change the background color to white without making any changes.

In addition, you can highlight words or lines from the Fingerprint CCO file in the images by clicking the CCO Words or CCO Lines icons.

Attention: Image rotation and rescan change the page image file only; associated data such as the page data file and CCO are not affected. After you rotate or rescan an image, the OCR results might no longer be valid and you must run OCR again. For example, you can create an alternative task profile that runs OCR after image rotations.

Restriction: When you use Internet Explorer 9 to import multiple pages, you must specify the number of pages to import from the selected directory. In Internet Explorer 10 or later, you can select specific pages to import from a directory by using the Shift or Ctrl keys.

Parent topic: [Administering Datacap Navigator](#)

Related tasks:

[Setting Datacap Navigator default page layouts](#)

Batch upload

After you run the Scan task to capture your batch, you run the Upload task to upload the images to the Datacap server. While this is a task that is automated by Rulerunner in a production environment, you can upload images manually in Datacap Navigator.

To find a batch more quickly, enter information from any Task List column in the Batch Filter field. For example, enter `pending` in the Batch Filter field to view only the jobs with a pending status.

To customize the columns of the table, click the user ID drop-down menu and select Change User Settings and click the Task List tab.

Tip: Click the Refresh button to update the Task List frequently because other operators might be changing the status of batches or jobs.

Parent topic: [Administering Datacap Navigator](#)

Page classification

Document assembly is usually an automatic process that runs in the background. However, in some cases the automatic assembly task does not correctly identify page types. When page types are incorrectly identified, you can use Classify to manually edit the type for each page and modify the batch structure.

You edit the page type and status of any page in the batch within the Batch Structure pane of the Classify task. Double-click a page in the Image Viewer pane to highlight that page in the Batch Structure. You can also use the Batch Structure pane to modify the document hierarchy.

Restriction: When you use Internet Explorer 9 to import multiple pages, you must specify the number of pages to import from the selected directory. In Internet Explorer 10 or later, you can select specific pages to import from a directory by using the Shift or Ctrl keys.

Parent topic: [Administering Datacap Navigator](#)

Batch processing (Task List)

You can monitor the status of batches and jobs for a specific task and run pending jobs from the Datacap Navigator Task List page.

During the data capture process, documents go through a workflow that consists of several discrete tasks such as scanning, upload, classification, and verification. Datacap uses a queuing mechanism to move batches of documents through the workflow.

You can run a task on a batch by clicking the batch in the Task List page and selecting one of the available actions.

When the Program setting of a scan task is set to `Multiple`, the task list displays batches for every station.

You can filter the batches by their string, numeric, or date columns. If there are more batches that has not been retrieved from the server, you have the option to continue the filtering on the server to bring back all the matches from the server

To customize the columns of the table, click the user ID drop-down menu and select `Change User Settings` and click the `Task List` tab.

Tip: Click the `Refresh` button to update the Task List frequently because other operators might be changing the status of batches or jobs.

To **search by date**, open Filter window, select a Date column, for example, `Job Start` from Column drop-down.

- To filter exact batch date, select `equals`
- To filter before a date, select `before`
- To filter after a date, select `after`
- To filter batches from a range of dates, select `from date and to date`.

Then click `Filter`. You can see all the matching batches.

Parent topic: [Administering Datacap Navigator](#)

Setting Datacap Navigator default page layouts

The `Classify`, `Verify`, and `Scan` pages contain widgets such as the image viewer, start panel, field details, and batch structure. You can set the default location of the widgets for specific tasks.

About this task

Datacap Navigator users can override the default layout by changing the user settings. From any Datacap Navigator view, click the user ID drop-down menu and select `Change User Settings`. On the `Settings` window, click the `Classify`, `Verify`, or `Scan` tab. Then, click the `Layout` tab and rearrange the widgets.

Procedure

To set the default layout of the `Classify`, `Verify`, and `Scan` pages:

1. Open the Administration View. In Datacap Navigator, click the Administration View icon, or enter the following URL in a browser:

```
https://server:port/context_root/?desktop=dcadmin
```

2. Click `Workflows` in the left pane.
3. Select a workflow in the right pane and click `Edit`.

4. Click the Jobs tab, select a job, and click Edit.
5. Click the Tasks tab, select a classify, verify, or scan task, and click Edit.
6. Click the Layout tab and set the default location of the widgets.
7. Click Save and Close.

Parent topic: [Administering Datacap Navigator](#)

Related information:

[Setting Datacap Navigator default layouts in Datacap Version 9.0 and Feature Packs 1 and 2](#)

Constructing a URL for Datacap Navigator

You can construct a URL that an external application can use to access Datacap Navigator features and actions. You can append parameters to the URL that run actions and open Datacap Navigator pages.

Before you begin

You must configure IBM® Content Navigator to support Secure Sockets Layer (SSL) communication when you use URLs to access Datacap Navigator features. For more information, see [Planning for Secure Sockets Layer \(SSL\)](#).

Procedure

To construct and use a URL to access Datacap Navigator features and actions:

1. Start with the following base URL:

```
https://server:port/context_root/?
```

Use the question mark (?) before you append any parameters.

2. Append the appropriate parameters to the base URL.

Separate each parameter-value pair with an ampersand (&).

The following lists describe the parameters that you can append to the URL, the appropriate format, and valid values for each parameter.

Running Datacap Navigator actions

You can add parameters to the URL to run existing actions or custom actions. The following table lists actions that are available on the Job Monitor page.

Table 1. Valid parameters and values for running actions.

Parameters	Values
------------	--------

Parameters	Values
dcAction	<p>editBatchPluginAction</p> <p>Specifies the edit batch action.</p> <p>Format: dcAction=editBatchPluginAction</p> <p>editJobPluginAction</p> <p>Specifies the edit job action.</p> <p>Format: dcAction=editJobPluginAction</p> <p>viewBatchHistoryPluginAction</p> <p>Specifies the view batch history action.</p> <p>Format: dcAction=viewBatchHistoryPluginAction</p> <p>viewBatchPropertyPluginAction</p> <p>Specifies the view batch property action.</p> <p>Format: dcAction=viewBatchPropertyPluginAction</p> <p>executeBatchPluginAction</p> <p>Specifies the start batch action.</p> <p>Format: dcAction=executeBatchPluginAction</p> <p>deleteBatchPluginAction</p> <p>Specifies the batch delete action.</p> <p>Format: dcAction=deleteBatchPluginAction</p>
queueID	<p>Specifies the queue ID of the action. The queueID parameter is not required if the action does not apply to a queue item.</p> <p>Format: queueID=<i>queue_ID</i></p>

Running shortcuts

You can run a Datacap Navigator shortcut. If you specify a job and task, the shortcut is run for the specified job and task only. Running a shortcut with a specified job and task is analogous to clicking a job under a shortcut in the Datacap Navigator shortcut pane.

Tip: To run a scan, specify a scan shortcut in the URL.

Table 2. Valid parameters and values for running Datacap Navigator shortcuts.

Parameters	Values
dcShortcut	<p>Specifies the name of the shortcut.</p> <p>Format: dcShortcut=<i>shortcut_name</i></p>
dcJob	<p>Specifies the name of the job.</p> <p>Format: dcJob=<i>job_name</i></p>

Parameters	Values
dcTask	Specifies the name of the task. Format: dcTask= <i>task_name</i>

Opening Datacap Navigator pages

You can open a specific page of Datacap Navigator by using the IBM Content Navigator feature parameter.

Table 3. Valid parameters and values for opening Datacap Navigator pages.

Parameters	Values
feature	DatacapMainFeature Opens the Datacap Navigator main page. Format: feature= <i>DatacapMainFeature</i> DatacapWebAdminFeature Opens the Datacap Navigator Administration View. Format: feature= <i>DatacapWebAdminFeature</i>

More IBM Content Navigator parameters are available. For example, you can use the sideChrome parameter to hide the Datacap Navigator side bar and banner. For more information, see [Constructing a URL for IBM Content Navigator](#).

Example

Open the Datacap Navigator main page

```
https://myserver.mycompany.com:myport/navigator/?desktop=datacap
&feature=DatacapMainFeature&sideChrome=0
```

Tip: To show Job Monitor only, hide the shortcut pane and quick launch pane in your desktop definition.

View the history of a batch with an associated queue ID

```
https://myserver.mycompany.com:myport/navigator/?desktop=datacap
&feature=DatacapMainFeature&sideChrome=0&dcAction=
viewBatchHistoryPluginAction&dcQueueID=3
```

View the history of a batch with an associated Station ID

You can view history of a batch with an associated Station ID. For example, if you used Station ID as *Lansing*, then after scanning, the scan batches show the Station ID as *Lansing*.

Run a queue

```
https://myserver.mycompany.com:myport/navigator/?desktop=datacap
&feature=DatacapMainFeature&sideChrome=0&dcAction=
executeBatchPluginAction&dcQueueID=3
```

Run a Verify shortcut with a specified job (Web job) and task (Verify)

```
https://myserver.mycompany.com:myport/navigator/?desktop=datacap
&feature=DatacapMainFeature&sideChrome=0&dcShortcut=
Verify&dcJob=Web%20Job&dcTask=Verify
```

Run a Verify shortcut without specifying a job and task

```
https://myserver.mycompany.com:myport/navigator/?desktop=datacap
&feature=DatacapMainFeature&sideChrome=0&dcShortcut=Verify
```

Authenticating without single sign-on (SSO)

If you are not using SSO, you can authenticate with an IBM Content Navigator web service call (jaxrs/logon) that has the following parameters in the body: desktop, userid, password, and contextPath. When the request succeeds, the logon session to the IBM Content Navigator server is returned. Then, you can use a URL to access Datacap batches in the authentication repository of that desktop.

Your Datacap application must be specified as the default authentication repository for your desktop. To handle cross-domain restrictions, you can put the logon result into an iFrame to know when the logon request completes and when you can access your Datacap batches. See the following example of a logon that uses a form submit request:

```
<form action="https://myserver:myport/navigator/jaxrs/logon" method="post"
target="output_frame">
  <input type="text" name="desktop" value="datacap" ></input>
  <input type="text" name="contextPath" value="/navigator" >
</input>
  <input type="text" name="userid" value="admin" ></input>
  <input type="text" name="password" value="admin" ></input>
  <input type="submit" value="Go to Navigator" />
</form>
```

Tip: If a logon window is displayed in Internet Explorer when you access Datacap Navigator with a URL, you can resolve this issue by adding the Datacap Navigator IP or domain address to your list of managed sites. In Internet Options, on the Privacy tab, click Sites. Then, in the Per Site Privacy Actions window, enter the Datacap Navigator IP or domain address in the Address of website box and click Allow. The following sample HTML code shows how you can log on and open Job Monitor without SSO. You can create a blank HTML file, paste the sample code into the file, edit as required, and then open the file in a browser.

```
<html>
  <header>
    <Script>
      window.onload = function() {
        var iFrame = document.getElementById("output_frame");
        iFrame.onload = function () {
          window.open("https://myserver:myport/navigator/?
desktop=datacap");
        };
      }
    </Script>
  </header>

  <body>
    <form action="https://myserver:myport/navigator/jaxrs/logon"
method="post"
target="output_frame">
      <input type="text" name="desktop" value="datacap"
></input>
      <input type="text" name="contextPath" value="/navigator"
></input>
      <input type="text" name="userid" value="myusername"
></input>
      <input type="text" name="password" value="mypassword"
></input>
      <input type="submit" value="Go to Navigator" />
    </form>
    <iframe name="output_frame"
src="/dcs/markdown/workspace/Transform/htmlout/0/com.ibm.dc.admin.doc/"
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.admin.doc_dcnav003_o
utput_frame" width="300"
height="300" >
```

```
</iframe>
</body>
</html>
```

Parent topic: [Administering Datacap Navigator](#)

External data services for Datacap Navigator

You can use the IBM® Content Navigator external data services (EDS) REST protocol to get data such as files or database tables from an external source. You can use the external data to customize field properties and manage property behavior in Datacap Navigator.

When you create an external data service, the data is integrated with Datacap field values and other field properties, and you continue to store and maintain the data only in the original, authoritative data source.

An advantage to using an external data service is that you do not need to modify the Datacap source code to customize field values and behavior. Upgrades and other major changes to Datacap do not affect the field data that is obtained by an external data service. The data and external data service are located separately from Datacap source code.

Where external data services can be implemented

You can implement external data services for the following fields in Datacap Navigator:

- Fields on the verify page
- Batch level fields in the start panel and batch editor panel
- Extra fields in the start panel and batch editor panel
- Batch attributes in the start panel and batch editor panel

You can use external data services to customize the following field properties and property behaviors:

Look up values in a database to create choice lists

Create choice lists by using existing data that is managed in a different content repository or data source outside of Datacap.

For example, you can use values in a file that is located and managed in an external server or repository.

Prefill properties

Specify prefilled properties and default values.

For example, you can prefill fields with custom default values that are based on a particular field ID, authenticated user, or the application.

Set DataType variable

Specify the DataType variable for a field so that you can specify different editors when you create custom panel. For more information about DataType variable, see [DataType](#).

Set minimum and maximum values

Specify an integer, float, or date to define the maximum or minimum value for a field.

Set read-only status

Set a field to be read-only.

For example, you might create a property that requires a particular value. To prevent users from entering a different value that could cause an error, you can specify the correct default value and make that property read-only.

Set required status

Set a property to be a required field. When you use this attribute on a property, an asterisk appears in the user interface to indicate that the field is required. Users cannot proceed from the page or dialog box unless the field contains a value.

Set hidden status

Hide a property from the user interface.

Show a custom message or provide assistance when users enter values into a property field.
Restriction: You cannot use custom validation for object properties, reference attributes, read-only properties, hidden properties, or search criteria.

For example, you might create a choice list that dynamically determines subsequent text input fields to present in a form. To hide a property that does not apply in a particular situation, you can use the hidden attribute.

Implement field validation and error checking

Show a custom message or provide assistance when users enter values into a property field.

Restriction: You cannot use custom validation for object properties, reference attributes, read-only properties, hidden properties, or search criteria.

EDS REST protocol specifications

Object types resource

The object type resource represents the names of the Datacap applications to get for the external data service. Example response for Datacap Navigator:

```
[
  {"symbolicName": "TravelDocs"},
  {"symbolicName": "FastClaim"}
]
```

For more information, see [Object types resource](#).

Particular object type resource

The particular object type resource represents the properties for which property values are obtained from an external data source. When the user scans a batch, edits a batch, or verifies a batch of a particular Datacap application, the EDS REST protocol uses the particular object type resource to obtain data for the corresponding Datacap application from the external data source.

Request modes

initialNewObject is supported.

Response payload

The following parameters are supported: choiceList, format, formatDescription, required, and value.

For more information, see [Particular object type resource](#).

Request modes

The following request modes are supported by Datacap Navigator:

Request mode	Datacap action
initialExistingObject	Create batch Get batch list Get page file Get data file
inProgressChanges	Get dependent attribute information Save page file Save data file
finalExistingObject	Save page file Save data file Save batch attributes

For more information, see [Request modes](#).

Sample external data service

A sample external data service and JSON files are installed by default in the C:\datacap\tmlweb.java\samples directory. To download the latest version of the sample external data service and JSON files, including all fixes and updates, see: [EDS sample for Datacap Navigator](#).

Important: The versions of the EDS sample and the Datacap Navigator plugin must match. When you install a new version of Datacap Navigator (for example, from a new Datacap release, feature pack, or fix pack), be sure to use the compatible version of the EDS sample that is installed in the \datacap\tmlweb.java\samples directory.

To deploy the EDS sample, see [Deploying the sample external data service](#).

In the sample external data service, the values for a choice list are used from an external data source.

The sample service is implemented against the TravelDocs sample application. You can use the sample service as an example and as a starting point for your own external data service. The sample service consists of a GetObjectTypes servlet, an UpdateObjectTypes servlet, the object types resource JSON file, JSON files as the data source, and the web.xml deployment file.

The application name is set as the objectType, and you can add a symbolicName for each field (page field, batch level field, or extra field) in the application. For more information, see the TravelDocs_PropertyData.json and ObjectTypes.json files that are provided with the sample service.

Modify the sample external data service as required for your application. Then, deploy the service in the web application server as follows.

1. Deploy the DatacapEDSService application into your application container.
2. Deploy edsPlugin into IBM Content Navigator. edsPlugin is a plug-in that is provided with IBM Content Navigator. Confirm that edsPlugin is enabled after deployment.
3. Edit edsPlugin. Point the edsPlugin configuration URL to the external data service, for example:
`http://IP_address:port/DatacapEDSService`.
4. Log out and log on to the desktop in which you want to use the external data service.

Parent topic: [Administering Datacap Navigator](#)

Customizing Job Monitor

You can customize the Datacap Navigator Job Monitor page by using external data services (EDS). For example, you can change cell values, styles, column names, and enable cells to show Dojo widgets.

Cell properties

You can change cell properties such as field values. For more information, see the sample external data service: [EDS sample for Datacap Navigator](#).

Column properties

You can change column properties such as the column name and style. You can define the style with a CSS snippet, for example:

```
background-color: #b0c4de;
```

Alternatively, you can use a JavaScript function to change column properties. Define a string that is evaluated as a function on the JavaScript side. Set the function to a variable named `func`, for example:

```
var func = function(cell){};
```

Then, define the column style as shown in the following example:

```
cell.put("style", "background-color: #b0c4de;");
cell.put("style", "var func = function(cell){var data = cell.rawData();
if(data==='aborted'){return 'background-color: #ff0000;'}};");
```

Showing Dojo widgets in cells

You can declare a widget by setting `widgetsInCell` to `true` and `decorator` to a function. For example, the following code displays the `ProgressBar` dijit in a cell:

```
cell.put("widgetsInCell", true);
cell.put("decorator", "progressBarDecorator");
```

`progressBarDecorator` is a JavaScript function that is used by `decorator` and can be defined in your IBM® Content Navigator plug-in. You can replace the existing `ProgressBar` widget with widgets that are defined in your plug-in, for example:

```
lang.setObject("progressBarDecorator", function(data, rowId, rowIndex) {
  return [
    "<div data-dojo-type='dijit.ProgressBar' data-dojo-props='maximum: 1' ",
    "class='gridxHasGridCellValue' style='width: 100%;'></div>"
  ].join(' ');
});
```

The Job Monitor GridX widget uses the `cells` object in the payload as a column structure definition. By default, the `cells` object does not include extra columns. When you update the properties of an extra column in EDS, you must add the column to the `cells` object manually. If you are working with a default column that is already defined in the `cells` object, simply update the column.

Parent topic: [Administering Datacap Navigator](#)

Related concepts:

[External data services for Datacap Navigator](#)

Datacap Navigator access

You can access Datacap Navigator in an administrator view or a non-administrator view.

URLs

In the following URLs, the *server* and *port* are the server and port where IBM® Content Navigator is deployed.

Non-administrator view	<code>http://server:port/navigator/?desktop=datacap</code>
Administrator view	<code>http://server:port/navigator/?desktop=dcadmin</code>

Parent topic: [Administering Datacap Navigator](#)

Datacap Desktop administration

You can configure the application tasks that Datacap Desktop runs and modify the way that those tasks are performed.

- [Creating and configuring a task to use with Datacap Desktop](#)
The Datacap Studio application wizard creates a workflow that includes the Main Job, a Fixup Job, and a Web Job. By default, the application wizard does not create a scan task that you can use for scanning hardcopy pages. To scan pages and create a batch with Datacap Desktop, you need to manually create a scan task.
- [Using command line parameters for Datacap Desktop](#)
You can use various command line parameters to launch applications using Datacap Desktop
- [Setting up a barcode type as a document separator](#)
You can set up your Datacap Desktop scan task with a barcode type to be used as a document separator with ISIS scanners.
- [Configuring the image selection mode for the Datacap Desktop Fixup task](#)
For the Fixup task, the default behavior of Datacap Desktop is to retain the last scanned page image and to automatically delete any previous page images. For TWAIN and ISIS scanned pages, you can override this default behavior so that you manually select which page image to retain.

Parent topic: [Administering](#)

Creating and configuring a task to use with Datacap Desktop

The Datacap Studio application wizard creates a workflow that includes the Main Job, a Fixup Job, and a Web Job. By default, the application wizard does not create a scan task that you can use for scanning hardcopy pages. To scan pages and create a batch with Datacap Desktop, you need to manually create a scan task.

Before you begin

If your 8.1 application was using DotEdit or DotScan, you must update the setup information in Datacap Web Client to use Datacap Desktop. If you do not see the setting options for Datacap Desktop, change the selected program, and then the setting options for Datacap Desktop are automatically available.

If the program that you want to use is already selected, change to another program, then select your original program, and press apply. For example, if your program is Rulerunner, change the program to Multiple, then select Rulerunner again and press Apply. The Datacap Desktop setup options are available.

About this task

The Main Job and the Web Job both contain sufficient tasks for processing a batch, from batch creation to exporting the batch. However, you might need to create a task that requires special handling, such as scanning hardcopy pages.

Procedure

Follow this procedure to create a task to use with Datacap Desktop.

1. In your web client, log on to the application that requires a Datacap Desktop task.
2. Select Workflow.
3. Select the job that contains the task, and click New.
Tip: You can modify an existing task to use Datacap Desktop by selecting the existing task in the job, and specifying Datacap Desktop as the program to use. See Step 6.
4. Enter a name for the new task, an appropriate description.
5. Select the values for these fields.

- a. Mode: Select one of these values according to your requirements:
 - Batch Creation: Select this mode for use with VScan or if you are creating a task to scan hardcopy documents from Datacap Desktop.
Important: A job can contain only one Batch Creation task. If the job that you are modifying already includes a batch creation task, you must remove that task.
 - Router: Select this mode if your new task routes the batch to a different task or job when the criteria of a condition are met. One example of a condition is a document integrity failure that requires a supervisor's intervention.
 - Normal: This mode is for all other tasks that are not used for Batch Creation or that do not require special handling.
 - b. Queue to: You can specify whether to queue the task to a user, a workstation, or both. If queuing is not a requirement, select None.
 - c. Store: For reporting purposes, you can specify whether to save task completion information that is based on a user, a workstation, or both. To reduce processing time, you can select None.
6. For the Program under the Parameters section, select Datacap Desktop.
 7. Click Create Setup, and then click Setup.
 8. Specify the values for the fields, if necessary. The correct settings for Datacap Desktop to call Rulerunner is for the key field to be Default and value field to be dcDTlib.rulerunner:B. In most cases, you can accept the defaults, unless you are creating a scan task.

If you are creating a scan task, update the Datacap Desktop section with the following settings:

- a. Add another set of key and value fields by clicking the plus sign (+).
Note: Only one key-value pair is needed for the scan tasks. The pair Page_TYPE and Panel Name is provided only for guidance, and can be replaced.
 - b. In the key field enter `Default`.
 - c. In the value field, enter `DotScanPanels.ISISScan:B` to use an ISIS scanner, `DotScanPanels.TWAINScan:B` to use a TWAIN scanner, or `DotScanPanels.VScan:B` to select images from disk files.
9. If necessary, select the new task and move it to the correct location in the job.
Important: A batch creation task must be the first task in a job.

Parent topic: [Datacap Desktop administration](#)

Using command line parameters for Datacap Desktop

You can use various command line parameters to launch applications using Datacap Desktop

Command line parameters for Datacap Desktop

You can use the following three command line parameters to launch applications using Datacap Desktop.

`-nta`

Use the Windows username as the user ID for login screen. This command launches the login dialog with the User field set to the current Windows login Id and the Password field disabled. If the login fails (after selecting an application), the login dialog appears with the Password field enabled.

Syntax: `C:\Datacap\DcDesktop\DcDesktop.exe -nta`

`-app:<appname>`

Log in to the specified application. This command launches the Datacap Desktop program and automatically selects the specified application.

Syntax: "C:\Datacap\DcDesktop\DcDesktop.exe -app:<ApplicationName>"

Example: "C:\Datacap\DcDesktop\DcDesktop.exe -app:TravelDocs"

In this example, the command launches the Datacap Desktop program and automatically selects the TravelDocs application.

-but:<button>

Run the specified shortcut on startup. This command launches the Datacap Desktop program and automatically selects the specified application with the specified shortcut.

Syntax: "C:\Datacap\DcDesktop\DcDesktop.exe -app:<ApplicationName> -but:<Shortcut>"

Example: "C:\Datacap\DcDesktop\DcDesktop.exe -app:TravelDocs -but:Verify"

In this example, the command launches the Datacap Desktop program and automatically selects the TravelDocs application with the Verify shortcut.

Parent topic: [Datacap Desktop administration](#)

Setting up a barcode type as a document separator

You can set up your Datacap Desktop scan task with a barcode type to be used as a document separator with ISIS scanners.

About this task

To set up a barcode type as a document separator, you must complete three steps. You must configure the Datacap Desktop scan task for document creation and set the DCO type panel. Then, scan a page with the barcode that you want to use as a document separator and identify the barcode name. The last step is to enter the barcode name as the Barcode type in the Datacap Desktop scan task setup.

Procedure

1. Configure the scan task in Datacap Web Client for documentation creation and set the DCO type panel.
 - a. In Datacap Web Client, go to Administrator > Workflow and select the Scan task.
 - b. Select Datacap Desktop from the Program menu.
 - c. Select Setup.
 - d. In Batch processing, select the Create document node check box.
 - e. In Datacap Desktop Bind DCO type to panel section, enter `DCO_Type` in the key field and `DLL.Panel` in the value field
 - f. Add another Desktop panel set of key and value fields. Enter the application name, such as `TravelDocs`, in the key field. Enter `DotScanPanels.ISISScan` in the value field.
2. Scan a page with the barcode that you want to use as a document separator and identify the barcode name.
 - a. Scan a page with the barcode that you want to use as a document separator. Submit and close that batch.
 - b. In the batch folder, open the Scan.xml file and find the `CodeNameX`, such as `Code39`.
3. In the Datacap Web Client, set up the Datacap Desktop scan task with the Barcode type.
 - a. In Datacap Web Client, go to Administrator > Workflow and select the Scan task
 - b. In the Barcode type field, enter the barcode name, such as `Code39`, that you identified in the Scan.xml file
4. Save this Datacap Desktop scan task setup.

Configuring the image selection mode for the Datacap Desktop Fixup task

For the Fixup task, the default behavior of Datacap Desktop is to retain the last scanned page image and to automatically delete any previous page images. For TWAIN and ISIS scanned pages, you can override this default behavior so that you manually select which page image to retain.

To enable manual selection mode, add the following line to the Datacap Desktop section of the Fixup task for one or more of the following files:

- VScan.set.xml
- fixup.set.xml

```
<V n="UndoableRescan" label="Undo-able Rescan"  
  tip="Specify whether rescanning is undo-able" type="checkbox">1</V>
```

Parent topic: [Datacap Desktop administration](#)

Datacap Application Copy Tool

The Datacap Application Copy Tool is a migration tool that you can use to copy and move a Datacap application. For example, you can move the application from a test environment to replace an existing application in a production environment.

Some Datacap applications use a settings.ini file that contains UNC paths and connection strings. The Datacap Application Copy Tool does not include this file when it copies the application files.

When you copy databases to another environment, the Engine database is usually not copied. You copy the Administration database and either create a new Engine database or reuse an existing Engine database schema.

When Datacap Application Copy Tool is used as a command-line interface, it is an automated alternative to the Datacap Studio Application wizard to the run the following tasks:

Move new data into an environment for the first time

Copies all of the application files and databases from a Datacap application in one environment and moves them into a new application in another environment. There are some files that are not copied, such as settings.ini, batches, input directories, output directories, and non-standard directories. You must verify and correct the connection string to the new application.

Creates a new index for the Administration database in the new application. For example, you can create the application in a development environment and move its application files and databases into a test environment for quality testing and verification. After the application is tested and ready for use, you can then move the entire application into a production environment.

To rename an existing application, you use the Application Wizard in Datacap Studio.

Update data in an existing environment

Copies updated application files and databases to an existing environment. The application in the existing environment is an older version of the application that is being copied. This older version is updated to match the application that you are copying. For example, you can change the document hierarchy and rules in an existing application and copy the updated application into another environment. You can move the entire application or only the parts of the application that you updated.

You can use this option to migrate your applications up to a new environment or to back to a previous environment.

Change the database provider on your system

Copies databases from an existing database provider and moves them to another database provider. For example, you can move your Datacap application databases from a Microsoft Access database in one environment to a DB2® database in another environment. The two environments must mirror each other to ensure the data that is passed between the database providers is synchronized.

To change the database provider, you must clear the Copy application files check box and manually enter the connection strings for the destination database. You do not have to enter the target path and application name. After the files are copied, you update the connections strings for the application to use the new database in Application Manager.

Back up the data on your system

Copies all of the application files and databases to a single folder to create a checkpoint or backup of your system.

- [Configure the connection strings](#)
You use connection strings to copy and move Datacap applications to a database in another environment. Connection strings contain the information that is needed by the database provider to establish a connection between the application and the database.
- [Datacap Application Copy Tool Command Line Interface](#)
By using the command line options for the Datacap Application Copy Tool, you can specify the Datacap application that you want to copy and move that application into another environment.
- [Datacap Application Copy Tool User Interface](#)
You can use the Datacap Application Copy Tool User Interface to move your Datacap applications and their databases into another environment.

Parent topic: [Administering](#)

Configure the connection strings

You use connection strings to copy and move Datacap applications to a database in another environment. Connection strings contain the information that is needed by the database provider to establish a connection between the application and the database.

When you connect your application to a database, ADO uses a database provider to make the connection to the database. The supported database providers for which you can configure connection strings are Microsoft Access, Microsoft SQL Server, Oracle, and DB2®. The old Datacap format connection strings are still supported to provide compatibility with an earlier version to your existing database connections.

Database providers make the connection to a database in different ways so you must write the connection string for the database provider that you want to use. For example, the database provider needs the address of the database server to connect to. You specify the address in the connection string to tell the database provider where to connect.

If you do not specify connection strings for an existing application, the database provider uses the connection strings that are currently configured on the application.

The following examples describe OLEDB connection strings for each of the supported database providers:

DB2 by using Standard Authentication

This example describes the primary database. The Data Source value is the database alias name that is cataloged from the Database Server that is installed on Datacap Server. The Hostname and Database values can be empty.

```
"Provider=IBMDADB2;Data Source=dcdb141;UID=*****;PWD=*****;  
CurrentSchema=db2admin;"
```

This example describes the secondary database, which already exists. The Hostname value is the IP address of the remote Database Server. The Database value is the name of the database that is installed on the Database Server. The DataSource value can be empty.

```
"Provider=IBMDADB2;Hostname={IP address};Data Source=;Database=dcdbNUI;  
Password=*****;User ID=*****;"
```

DB2 by using Windows Authentication

N/A

Microsoft Access by using Standard or Windows Authentication

```
"Provider=microsoft.jet.oledb.4.0;data source=C:\Datacap\MyApp\MyAppadm.mdb;  
persist security info=false;"
```

SQL Server by using Standard Authentication

```
"Provider=sqloledb;data source=myServerAddress;Initial Catalog= myDataBase;  
User Id=myUsername;Password=myPassword;"
```

SQL Server by using Windows Authentication

```
"Provider=sqloledb;data source=myServerAddress;Initial Catalog= myDataBase;  
Integrated Security=SSPI;"
```

Oracle by using Standard Authentication

```
Provider=OraOLEDB.Oracle;Data Source=MyOracleDB;User Id=myUsername;  
Password=myPassword;
```

Oracle by using Windows Authentication

```
Provider=OraOLEDB.Oracle;Data Source=MyOracleDB;OSAuthent=1;
```

Parent topic: [Datacap Application Copy Tool](#)

Related information:

[Planning authentication for your Datacap system](#)

Datacap Application Copy Tool Command Line Interface

By using the command line options for the Datacap Application Copy Tool, you can specify the Datacap application that you want to copy and move that application into another environment.

Command line syntax

The following Backus Normal Form (BNF) describes the command line syntax that is used by the Datacap Application Copy Tool to copy and move applications and databases.

```
<syntax> :=<source> <dest> [<app-status>] [<copy-type>*] [<keep-type>*] [<clear>*]  
[<reset-user>] [-s] [-i|-c]  
[-s] : = silent (use the Command Line Interface)  
[-i] : = interactive (display the UI)  
<source> :=-from [-an <app>] [-af <folder>] [-adb <DB>] [-fdb <DB>] [-edb <DB>]  
<app-type>  
<dest> :=-to [-an <app>] [-af <folder>] [-adb <DB>] [-fdb <DB>] [-edb <DB>] <app-  
type>
```

```
<app-type> :=-how <online|datafile>
<app-status> :=-oapp <new|update>
<copy-type> :=-copy <users|roles|workflow|rules|dco>
<keep-type> :=-keep <users|workflow>
<clear> :=-clear <audit|debug|batches|stats|fpstats>
<reset-user> :-reset <userid>
```

Commands

The Datacap Application Copy Tool uses the `-from` and `-to` parameters to specify the source and destination applications.

`-from`

Specifies the source application that you want to copy

`-to`

Specifies the destination application that you want to create or update

These two parameters introduce specifications for source and destination, such as the application name and database connections. For example, the following command, with appropriate values entered for the italicized items, copies a source application to the destination environment by using the online migration method.

```
DAppCopy -from -an Application -how online -to -an NewApplication
        -af NewApplicationFolder -how online -oapp new
```

Command-line options

The following options apply to the source environment when they are preceded by the `-from` parameter. These options apply to the destination environment when they are preceded by the `-to` command.

`-an Application`

The name of the application that you want to move

If the application exists, you can specify the *Application* name only. Omit the *Folder* and *DB* parameters.

`-af Folder`

Following the `-from` parameter, the path of the folder in the source environment from which you want to copy the application.

Following the `-to` parameter, the path of the folder in the destination environment into which you want to move the application.

`-adb DB`

The connection string to the Administration database of the source or destination application that you want to move. Use Microsoft OLEDB Connection syntax in the connection string.

`-edb DB`

The connection string to the Engine database of the source or destination application that you want to move. Use Microsoft OLEDB Connection syntax in the connection string.

`-fdb DB`

The connection string to the Fingerprint database of the source or destination application that you want to move. Use Microsoft OLEDB Connection syntax in the connection string.

`-how method`

The method that you want to use to access the source or destination application:

- `online`: clone the application from the source environment or copy it directly to the destination environment.

- **datafile:** copy the files and databases of the application to portable data files, or copy portable data files to files and databases in the destination environment.
- **backup:** copy all of the application files and databases to a single folder to create a backup for comparison purposes.

The following options do not apply to the source environment. They apply to the destination environment when they are used with the `-to` parameter.

-keep *users|workflow*

Preserves the users or workflow in the destination Administration database. To preserve users and workflows in the destination Administration database, use the `-keep` option twice. For example, enter `-keep users -keep workflow`.

-clear *audit|batches|debug*

- **audit:** Removes all records from the audit table in the destination Administration database.
- **batches:** Removes all batches from the destination Engine database.
- **debug:** Removes all debug records from the destination Engine database.

To remove multiple items, repeat the `-clear` option for each item that you want to remove. For example, enter `-clear audit -clear batches -clear debug`.

-reset *userid*

Resets the specified user's password to *admin* for the destination database. The user can then log in to the destination database even if the encryption keys do not match in the destination environment. The user is expected to immediately change the password to a more secure password.

-oapp *new|update*

The type of migration that you want to run:

- **new:** create a new application in the destination environment with the application files and databases from the source environment.
- **update:** update an existing application in the destination environment with the updated application files and databases from the source environment.

- [Application migration options](#)

You can migrate applications and databases on a computer that has access to both the source and target environments. You can also use multiple computers that each have access to one of these environments.

- [Moving an application into a new environment](#)

You can copy all of the application files and databases from a Datacap application in one environment and move them into a new application in another environment.

- [Updating an application in an existing environment](#)

You can update application files and databases move either the entire application or the updates to an existing environment.

- [Changing the database provider](#)

You copy the Datacap application databases that exist in a database provider and move them to another database provider. For example, you can move application databases from a Microsoft Access database in one environment to a DB2® database in another environment.

- [Migrating a single database to another database provider](#)

You can copy a single Datacap application database and migrate to use another database provider.

Parent topic: [Datacap Application Copy Tool](#)

Application migration options

You can migrate applications and databases on a computer that has access to both the source and target environments. You can also use multiple computers that each have access to one of these environments.

The following migration options clone the application files and databases of an existing application in the source environment and copy the application to a specified location in the destination environment.

- **Online:** Runs a one-step process in which you use a computer that must have access to both the source environment and the destination environment.
- **Datafile:** Runs a two-step process in which you use a computer with access to the source environment to create portable data files. The data files contain the application files and databases that you want to copy. You move the portable data files to a computer with access to the destination environment. You run Datacap Application Copy Tool on this computer to move the application files and databases into the destination environment. If both computers can access the portable data files, you do not have to physically move these files to the second computer.

The following use case examples on different databases to show the options to use for an Online migration and a Datafile migration.

Online migration

In this Microsoft SQL Server example, you clone a new APT application from a database in the source environment and copy it to a specified location in the destination environment. The computer must have access to both environments.

```
DAppCopy -from -an APT -af C:\Datacap\APT
-adb "Provider=SQLOLEDB;Data Source=localhost;Initial Catalog=APTAdm;
Integrated Security=SSPI;"
-edb "Provider=SQLOLEDB;Data Source=localhost;Initial Catalog=APTEng;
Integrated Security=SSPI;"
-fdb "Provider=SQLOLEDB;Data Source=localhost;Initial Catalog=APTFFP;
Integrated Security=SSPI;" -how online
-to -an APT -af C:\Datacap\APT
-adb "Provider=SQLOLEDB;Data Source=localhost;Initial Catalog=APTAdm;
Integrated Security=SSPI;"
-edb "Provider=SQLOLEDB;Data Source=localhost;Initial Catalog=APTEng;
Integrated Security=SSPI;"
-fdb "Provider=SQLOLEDB;Data Source=localhost; Initial Catalog=APTFFP;
Integrated Security=SSPI;"
-how online -oapp new
```

Datafile migration

In this Microsoft Access example, you copy the TravelDocs application in the source environment to portable data files on flash drive Z:

```
DAppCopy -from -an TravelDocs -af C:\Datacap\TravelDocs
-adb "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Datacap\TravelDocs\
TravelDocsAdm.mdb"
-edb "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Datacap\TravelDocs\
TravelDocs Eng.mdb"
-how online
-to -an PortApp -af Z:\PortApp
-adb "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=Z:\PortApp\DBs\
PortAdm.mdb"
-edb "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=Z:\PortApp\DBs\
PortEng.mdb"
-clear batches -how datafile -oapp new
```

You can simplify this datafile migration, for example:

```
DAppCopy -from -an TravelDocs -how online -to -af C:\temp\TravelDocsSnapshot
-how datafile
```

In this DB2® example, you copy the application from the portable data files on flash drive Z: to a new TravelDocs application in the destination environment.

```
DAppCopy -from -af c:\TravelDocsSnapshot -how datafile
-to -an TravelDocs2 -af \\SomeFolder\Datacap\TravelDocs
-adb "Provider=IBMDADB2;Hostname=DB2Instance;Database=TravelDocAdm;
Integrated Security=SSPI;"
-edb "Provider=IBMDADB2;Hostname=DB2Instance; Database=TravelDocEng;
Integrated Security=SSPI;"
-fdb "Provider=IBMDADB2; Hostname=DB2Instance;Database=TravelDocFP;
Integrated Security=SSPI;" -how online -oapp new
```

The Datacap Application Copy Tool creates a troubleshooting log file in %temp%\dappcopy.log, where %temp% is a Windows shortcut to the TEMP directory. For example, the TEMP directory might be C:\Users\USERNAME\AppData\Local\.

Parent topic: [Datacap Application Copy Tool Command Line Interface](#)

Moving an application into a new environment

You can copy all of the application files and databases from a Datacap application in one environment and move them into a new application in another environment.

About this task

When Datacap Application Copy Tool creates a new application, the copy process updates the environmental information in Datacap Application Manager (.app) for known values. These values include the Administration, Engine, and Fingerprint databases, and file paths that are in the application folder. Other environmental values, such as encrypted values, export and lookup database connections, and file paths that are not in the application folder, are emptied in the new application and must be manually updated by using Datacap Application Manager.

You can create an application in a development environment and move its application files and databases into a test environment for quality testing and verification. After the application is tested and ready for use, you can then move the entire application into a production environment.

When you move an application, the following files in the application folder are copied:

- The dco.xml and *.app files
- Folders with names that start with dco_
- Fingerprint files (optionally)

The Administration database contains workflows, users and groups, and authorization information.

The Fingerprint folder and database can be excluded if Fingerprint learning is enabled. If the destination environment has fewer fingerprints, you can use the Fingerprint Maintenance Tool.

In the Administration database, the audit table and Engine database table are cleared.

Procedure

To move an application into a new environment:

1. At the prompt, type `DAppCopy` followed by these commands all on one line.
2. In the `-from` command, specify the following options for the source environment.
 - a. `-an`: the name of the Datacap application database to copy.

- b. `-af`: the path name of the folder from which to copy the Datacap application.
 - c. `-edb`, `-adb`, `-fdb`: the connection strings that are used by the application database.
 - d. `-how`: the migration option to use as `-online` or `-datafile`.
 - e. `-oapp`: the type of migration to run as `new`.
3. In the `-to` command, you specify the same options for the destination environment.
Important: For a new application in the destination environment that uses a non-Microsoft Access database, you must specify the database provider in connection strings for the application databases. If the database provider is not specified, the databases are created in Microsoft Access.
 4. Batches within an application are not copied to the new environment. If you want to copy existing batches you must copy the batches folder. If the batch folder path is different in the new environment, you must update copied batch paths in the Engine database. Use a database tool to update the `PB_BATCHDIR` column values in the `TMBATCH` table to match the file structure on your target system, after migration is done.

Example

In this example new TravelDocs applications are moved from the source environment to the destination environment by using the online migration option. Both environments are using the DB2® database provider.

```
DAppCopy -from -an TravelDocs -af C:\Datacap\TravelDocs
-adb "Provider=IBMDADB2;Data Source=localhost;Initial Catalog=TravelDocAdm;"
-edb "Provider=IBMDADB2;Data Source=localhost;Initial Catalog=TravelDocEng;"
-fdb "Provider=IBMDADB2;Data Source=localhost;Initial Catalog=TravelDocFP;"
-how online -oapp new
-to -an TravelDocs -af C:\Datacap\TravelDocs
-adb "Provider=IBMDADB2;Data Source=localhost;Initial Catalog=TravelDocAdm;"
-edb "Provider=IBMDADB2;Data Source=localhost;Initial Catalog=TravelDocEng;"
-fdb "Provider=IBMDADB2;Data Source=localhost; Initial Catalog=TravelDocFP;"
-how online -oapp new
```

Parent topic: [Datacap Application Copy Tool Command Line Interface](#)

Related reference:

[Datacap Application Copy Tool Command Line Interface](#)
[Application migration options](#)

Updating an application in an existing environment

You can update application files and databases move either the entire application or the updates to an existing environment.

About this task

When Datacap Application Copy Tool updates an existing application, the destination `.app` file is not changed. Any keys that are added to Datacap Application Manager in the source application environment must be added to the destination application environment.

When you update an existing application, you can copy the actors, roles, and workflows that are associated with the application. The actors, roles, and workflows must be the same for the source and destination applications. The copy process might fail, if you copy only the workflows.

Procedure

To move an application into a new environment:

1. At the prompt, type `DAppCopy` followed by these commands all on one line.

2. In the `-from` command, specify the following options for the source environment.
 - a. `-an`: the name of the Datacap application database to copy.
 - b. `-af`: the path name of the folder from which to copy the Datacap application.
 - c. `-edb`, `-adb`, `-fdb`: the connection strings that are used by the application database.
 - d. `-how`: the migration option to use as `online` or `datafile`.
 - e. `-oapp`: the type of migration to run as `update`.
3. In the `-to` command, you specify the same options for the destination environment.

Example

In this example updated Medical Claims applications are moved from the source environment to the destination environment by using the online migration option. Both environments are using the DB2® database provider.

```
DAppCopy -from -an MClaims -af \\Development\Datacap\MClaims
-adb "Provider=IBMDADB2;Hostname=Devhost;Database=MClaimsAdm;"
-edb "Provider=IBMDADB2;Hostname=Devhost;Database=MClaimsEng;"
-fdb "Provider=IBMDADB2;Hostname=Devhost;Database=MClaimsFP;"
-how online
-to -an MClaims -af \\Production\Datacap\MClaims
-adb "Provider=IBMDADB2;Hostname=Prodhost;Database=MClaimsAdm;"
-edb "Provider=IBMDADB2;Hostname=Prodhost;Database=MClaimsEng;"
-fdb "Provider=IBMDADB2;Hostname=Prodhost;Database=MClaimsFP;"
-how online -oapp update
```

Parent topic: [Datacap Application Copy Tool Command Line Interface](#)

Related reference:

[Datacap Application Copy Tool Command Line Interface](#)

[Application migration options](#)

Changing the database provider

You copy the Datacap application databases that exist in a database provider and move them to another database provider. For example, you can move application databases from a Microsoft Access database in one environment to a DB2® database in another environment.

About this task

For any database except Microsoft Access, the destination database must be installed and able to run the scripts that create the database schema. When updating an existing application, the updated application must be the same application as the existing application. For example, you cannot update a TravelDocs application with an Accounts Payable application.

Procedure

To change the database provider:

1. At the prompt, type `DAppCopy` followed by these commands all on one line.
2. In the `-from` command, specify the following options for the source environment.
 - a. `-an`: the name of the Datacap application database to copy. Skip this step if you want to copy only the database and not the application files.
 - b. `-af`: the path name of the folder from which to copy the Datacap application.
 - c. `-edb`, `-adb`, `-fdb`: The connection strings that are used by the application database.
 - d. `-how`: the migration option to use as `-online` or `-datafile`.

3. In the `-to` command, specify the same options for the destination environment. Use the connection strings for the new database provider.

Example

In this example APT application databases are moved from Microsoft SQL Server in the source environment to DB2 in the destination environment by using the online migration option.

```
DAppCopy -from -an APT -af c:\Datacap\APTSQL
-adb "Provider=SQLOLEDB;Data Source=localhost;Initial Catalog=APTAdm;"
-edb "Provider=SQLOLEDB;Data Source=localhost;Initial Catalog=APTEng;"
-fdb "Provider=SQLOLEDB;Data Source=localhost;Initial Catalog=APTFP;"
-how online
-to -af C:\Datacap\APTDB2
-adb
"Provider=IBMDADB2;Server=MyDB2:5000;Database=APTAdm;UID=myUserName;PWD=myPassword;"
-edb
"Provider=IBMDADB2;Server=MyDB2:5000;Database=APTEng;UID=myUserName;PWD=myPassword;"
-fdb
"Provider=IBMDADB2;Server=MyDB2:5000;Database=APTFP;UID=myUserName;PWD=myPassword;"
-how onlineDAppCopy -from -an APT -af C:\Datacap\APT
```

Parent topic: [Datacap Application Copy Tool Command Line Interface](#)

Related reference:

[Datacap Application Copy Tool Command Line Interface](#)

[Application migration options](#)

Migrating a single database to another database provider

You can copy a single Datacap application database and migrate to use another database provider.

About this task

You can use the Datacap Application Copy Tool to run your existing Datacap application databases on another database product. For example, you might be running a Microsoft SQL Server database and you want to change to a DB2® database.

To migrate a database to another database provider, you specify only the source and destination database information on the command line.

Procedure

To migrate a database to another database provider:

1. At the prompt, type `DAppCopy` followed by these commands all on one line.
2. In the `-from` command, specify `-edb`, `-adb`, or `-fdb` and the connection information for the source database.
3. In the `-to` command, you specify the same information for the destination database.

Example

In this example a Datacap Administration database is migrated from Microsoft SQL Server to DB2.

```
DAppCopy -from -adb "Provider=SQLOLEDB;Data Source=localhost;Initial
Catalog=APTAdm;"
-to -adb
```

```
"Provider=IBMDADB2DB;Server=MyDB2:5000;Database=APTAdm;UID=myUserName;PWD=myPassword"
```

Parent topic: [Datacap Application Copy Tool Command Line Interface](#)

Datacap Application Copy Tool User Interface

You can use the Datacap Application Copy Tool User Interface to move your Datacap applications and their databases into another environment.

Procedure

To copy applications and databases on the Datacap Application Copy Tool User Interface:

1. In the Start menu, click All Programs > IBM Datacap Developer Tools > Datacap Application Copy Tool.
2. In the Copy from pane, specify the following information.
 - a. Select the location of the Datacap application and its databases.
 - b. Select the name of the application to move.
 - c. Optional: To reset a user's password to *admin* for the destination database, select Reset password for user and enter the ID for the user. The user can then log in to the destination database even if the encryption keys do not match in the destination environment. The user is expected to immediately change the password to a more secure password.
 - d. Enter the connection strings to the Administration and Fingerprint application databases that you want to move. Use Microsoft OLEDB Connection syntax in all of the connection strings.
3. In the Copy to pane, specify the following information if you are copying application files:
 - a. Select the name of the application to move or create a new application name.
 - b. Select the location where you want to copy the application.
 - c. Check Copy application files.
 - d. Check the items that you want to copy with the application files; Document Hierarchy, Rules.
 - e. Check the existing data that you want to save on the destination Administration database; Workflows, User and Groups.
 - f. Check Clear Audit table if you want to remove all of the existing records from the audit table in the Administration database.
 - g. Enter the connection string to the destination Administration database. Use Microsoft OLEDB Connection syntax in all of the connection strings. The connection strings must contain login and password credentials. They cannot contain the asterisks, which result from cutting and pasting a connection string from Datacap Application Manager.
 - h. Check Fingerprints if you want to save the existing fingerprints in the Fingerprint database.
 - i. Enter the connection string to the destination Fingerprint database. The connection strings must contain login and password credentials.
 - j. Check Clear Engine Database if you want to remove the existing batches from the Engine database.
 - k. Enter the connection string to the destination Engine database. The connection strings must contain login and password credentials.
4. Click OK to copy the application files and databases from the source database to the destination database.
5. Press Exit to shut down the Datacap Application Copy Tool.

Parent topic: [Datacap Application Copy Tool](#)

Related concepts:

[Configure the connection strings](#)

Monitoring system performance with IBM System Dashboard for Enterprise Content Management

You can monitor Datacap system performance through the IBM® System Dashboard for Enterprise Content Management.

You must download the dashboard software separately and install it in your environment, along with a supported version of Java™. You can then enable the associated dashboard listener component that is installed automatically with Datacap. Ensure that the dashboard product is not collocated on a computer with the Datacap components that you want to monitor.

Tip: The System Dashboard for Datacap provides a simple interface that you can use to view events. The license for IBM System Dashboard is included with all Datacap licenses. However, for additional functionality, you can purchase IBM ECM System Monitor under a separate license.

The dashboard listeners that are installed with the Datacap software provide counters to monitor client login activity, server requests, database actions, batch tasks, and file access. You can monitor actions for the Datacap Server, Rulerunner, and Datacap Web Client components.

By default, Datacap Server and Rulerunner are always available for dashboard monitoring. Datacap Web Client is also installed to be available for system monitoring, but you have the added ability to disable and re-enable the associated listeners through a setting in the server.ini file.

Enabling dashboard listeners

To enable or disable dashboard listeners for monitoring Datacap events, enter a value of 1 (enable) or 0 (disable) for the InformPCH parameter in the General section of the *install_path\datacap\tmlweb.net\server.ini* file. The server.ini file is located on the affected Datacap Web Client server where you want system activities monitored.

Datacap Server events

In the dashboard, you can monitor the Datacap Server events. All time-related counters are in nanoseconds.

The dashboard outputs Datacap Server information in a hierarchical presentation. For example:

```
Batches
Batches/Created
Batches/Created/Create time
Batches/Grabbed
Batches/Grabbed/Grab time
Batches/Released
Batches/Released/Release time
...
```

Table 1. Client actions for Datacap Server

Event	Description
Connect	A new client connected.
Login	A client logged in.
Logoff	A client logged off.
Disconnect	A client disconnected.

Table 2. OLEDB database events for Datacap Server

Event	Description
Open	A new connection to a database was created through OLEDB.
Time to open	The time it took to create a database connection through OLEDB.
Close	A connection to a database through OLEDB was closed.
Execute	An execute of an SQL statement was done through OLEDB, as in, any SQL statement that includes SELECT.
Execute time	The time it took for an execute of an SQL to be done through OLEDB, as in, any SQL statement other than SELECT.
Open recordset time	The time it took to open a recordset through OLEDB, as in, execute SELECT.

Table 3. ADO database events for Datacap Server

Event	Description
Open	A new connection to a database was created through ADO.
Time to open	The time it took to create a database connection through ADO.
Close	A connection to a database through ADO was closed.
Execute	An execute of an SQL statement was done through ADO, as in, any SQL statement that includes SELECT.
Execute time	The time it took for an execute of an SQL to be done through ADO, as in, any SQL statement other than SELECT.
Open recordset time	The time it took to open a recordset through ADO, as in, execute SELECT.
selectXML time	The time it took to open a recordset through ADO and return results in XML format.
atomic selectXML	An atomic selectXML action was done through ADO. This action creates a connection to a database, executes a SELECT SQL statement, formats the resulting recordset in XML, and closes the connection to the database.
atomic selectXML time	The time it took to do an atomic selectXML action through ADO.

Table 4. Batch events for Datacap Server

Event	Description
Created	A new batch was created.
Create time	The time that was required to create a batch.
Grabbed	A batch was grabbed by a client for processing.
Released	A batch was released by a client after processing.

Table 5. Files events for Datacap Server

Event	Description
Opened	A file was opened for a client.

Event	Description
Closed	A file was closed by a client.
IO	A file system input/output was done on a file that was requested by a client.
Time to open	The time that was required to open a file for a client.
Time to read	The time that was required to read a file contents for a client.
Time to write	The time that was required to save data to a file for a client.
Time to close	The time that was required to close a file for a client.

Rulerunner events

The Rulerunner events can also be monitored in the dashboard. All time-related counters are in nanoseconds. Some of the event counters are fixed, and some are dynamically created. Each configured *thread* creates a separate executable process of Rulerunner (RRProcessor.exe), and each process has an associated dashboard listener. Each listener has the same name, Rulerunner:, but with a unique ID number appended to it, such as Rulerunner::49205. Every event begins with the node Thread0. Two categorical nodes are always displayed: Applications and General.

Important: Rulerunner threads are actually separate processes. The initial node *Thread0* is a historical artifact and can be ignored.

The dashboard outputs Rulerunner information in a hierarchical presentation, for example:

```
Thread0
Thread0/Applications
Thread0/Applications/1040EZ
Thread0/Applications/1040EZ/Query Application Service
Thread0/Applications/1040EZ/Query Application Service/Duration
Thread0/General
Thread0/General/No pending batches
```

Under the Applications node, a subnode is displayed for each application that is configured, for example, TravelDocs, 1040EZ, or Datacap Accounts Payable (APT). Under each application node is a Main Job and a Query Application Service counter. Individual application-specific tasks are noted under the Main Job node.

Table 6. Application events for each Rulerunner process

Event	Description
Applications	A listener node under which all configured applications are displayed.
<i>application_name</i>	A listener node whose name is a configured application, for example, TravelDocs.
Main Job	The listener node under which typical application-specific tasks are listed.
<i>task_name</i>	A listener node whose name is a configured task, for example, PageID, Profiler, and Vscan.
Batch grabbed	The number of batches that were grabbed by an application for processing a task.
Batch released	A listener node that represents the number of batches that were released by an application after processing a task.
<i>batch_status</i>	A sublevel counter of Batch released, whose name shows the status of each released batch, for example, pending, aborted, hold, offline.

Event	Description
Batch statistics updated	The average time that was required to update the statistics through a SQL request.
Running RRS	The average time that was required for the Rulerunner Service to complete its running of rules.
Query Application Service	A counter that contains the total number of requests to Datacap Application Manager. Under this node is an accumulator that contains the average time spent to request information from Datacap Application Manager.

Table 7. General events for each Rulerunner process

Event	Description
Disconnected All	All Rulerunner threads disconnected.
Disconnected from TM server	A Rulerunner thread has disconnected from the Datacap server.
Logged off	A Rulerunner user logged off.

Datacap Web Client events

In the dashboard, you can monitor the Datacap Web Client events. All time-related counters are in nanoseconds.

The dashboard outputs Datacap Web Client information in a hierarchical presentation, for example:

```

application/Hits//tmweb.net/Task/gstopb.aspx
application/Hits//tmweb.net/Task/uplbfcl.aspx
application/Hits//tmweb.net/Task/utility.aspx
application/Hits//tmweb.net/Task/vscancl.aspx
application/Hits//tmweb.net/buttons.aspx
application/Hits//tmweb.net/jmonitor.aspx
application/Hits//tmweb.net/wflow.aspx
Clients
Clients/application/TMLogin
Clients/application/TMLogoff
...

```

Table 8. Application events for Datacap Web Client

Event	Description
Application	The application that was run.
Bits	The functions that were run and the time spent for each.
Get Buttons	The time that was required to present a URL for the user interface to present the user with list of buttons, shortcuts, or similar elements.
Stop Batch	The time that was required to stop a batch process.
TMLogin	A user action to log into Datacap Web Client.
TMLogoff	A user action to log out of Datacap Web Client.

Event	Description
Tricky Select XML	The time that was required to display database information in the user interface by using an internal database SQL select action, for example, for data lookup.
Hits	A designated number of bits that represents a user request in the user interface, for example, a request for a particular URL to display.

Table 9. Client events for Datacap Web Client

Event	Description
Clients	A node that represents a set of Datacap Web Client users.
Application	The application that a user is running, for example, Datacap Accounts Payable.
TMLogin	The time that was required to log into Datacap server.
TMLogoff	The time that was required to log out of Datacap server.
Session End	The time that was required to end a Datacap Web Client session.
Session Start	The time that was required to start a Datacap Web Client session.

Parent topic: [Administering](#)

Related information:

[IBM FileNet P8 system overview](#)

[IBM System Dashboard for Enterprise Content Management Users Guide](#)

Shutting down Datacap for maintenance

You must stop the Datacap software applications to perform system-wide maintenance, back up your environment, install new Datacap software, or upgrade or remove existing Datacap software.

About this task

Before you stop and restart Datacap, make a note of the Datacap software components that are running on each machine. Whether you are running the software on a single machine configuration or a Datacap network configuration, you must stop and restart the Datacap software in the appropriate sequence.

Procedure

To stop and restart Datacap software:

1. Shut down all of the Datacap client software, including Datacap Desktop, Datacap Studio, FastDoc, Maintenance Manager, Datacap Web Client, Datacap Navigator, the Datacap Web Client Upload Service, and the Rulerunner Service.
2. Shut down all of the web services such as Datacap Web Client, Report Viewer, Fingerprint Service, and Datacap Web Services.
3. Shut down the Datacap Server service.
4. Stop the databases.
5. Perform these steps in reverse order to restart Datacap.

Parent topic: [Administering](#)

Related concepts:

[AutoDelete batches with Datacap Maintenance Manager](#)

FastDoc maintenance

FastDoc provides maintenance features that manage the input files, folders, and files that are associated with completed and unfinished batches that can accumulate over time.

To prevent input files from accumulating, use the Delete Images setting on the Scan Settings panel to delete the original input files automatically from disk after image recognition is done.

To remove folders and files that are associated with completed and unfinished batches, FastDoc provides two functions that free up disk space.

- Purge finished batches
- Delete selected batch

In addition, FastDoc provides statistics about the processing that was done each day.

- [Purging finished batches](#)
Files that contain processing information that is associated with exported batches accumulate in your FastDoc Batches folder. When a batch is exported successfully and the processing files are no longer needed, you can purge them.
- [Deleting selected batches](#)
Unfinished batches can accumulate in your FastDoc Batches folder. If you are not going to complete the processing of an unfinished batch, you can delete the batch to free up disk space.
- [Viewing daily batch statistics](#)
FastDoc creates an XML file every day that contains daily batch statistics for completed Scan processes and the Export processes. The file name is the processing date in the format MM_DD_YYYY.xml.
- [Determine the original file name of a pre-scanned image](#)
When you process pre-scanned images that are produced on a system outside of FastDoc. You might need to know the FastDoc file name and the file name that is given to the image by the originating system.

Parent topic: [Administering](#)

Purging finished batches

Files that contain processing information that is associated with exported batches accumulate in your FastDoc Batches folder. When a batch is exported successfully and the processing files are no longer needed, you can purge them.

About this task

Purging files does not affect the exported image or TXT files and purging periodically is a good idea to free up disk space.

Procedure

To purge finished batches:

1. Start FastDoc.
2. In the Select a Batch window, with a count of the open and finished existing batches.
3. Click Purge finished batches. A message box displays the number of batches that are purged and you are asked to confirm this action.
4. Click Yes. A message box confirms the number of batches that were purged.

5. Click OK to close the message box. The batch folders were purged.

Parent topic: [FastDoc maintenance](#)

Deleting selected batches

Unfinished batches can accumulate in your FastDoc Batches folder. If you are not going to complete the processing of an unfinished batch, you can delete the batch to free up disk space.

Procedure

To delete selected batches:

1. Start FastDoc.
2. On the Select a Batch window, select the batch that you want to delete.
3. Click Delete selected batch. A message box displays the batch number of the batch you selected and you are prompted to confirm the deletion.
4. Click Yes. The message box closes and the batch is no longer displayed on the Select a Batch window.

Parent topic: [FastDoc maintenance](#)

Viewing daily batch statistics

FastDoc creates an XML file every day that contains daily batch statistics for completed Scan processes and the Export processes. The file name is the processing date in the format MM_DD_YYYY.xml.

Procedure

To view the daily statistics:

1. Start Microsoft Excel and select File > Open.
2. In Datacap mode, the daily statistics are stored in the Engine database.
3. In Local mode, go to the C:\Datacap\FastDoc\statistics folder and open the XML file as an XML list.
4. Review the information in the following columns.

Column Heading	Description
id	Batch ID
batch_created	Batch creation date/time stamp
scan_user	User ID and domain of the user who is running the scan process
scan_station	Name of computer on which scan was run
documents_scanned	Number of documents scanned
pages_scanned	Number of pages scanned
batch_exported	Date/time stamp of export
verify_user	User ID and domain of the user who is running verify process
verify_station	Name of computer on which verify was run
documents_exported	Number of documents exported
pages_exported	Number of pages exported

Determine the original file name of a pre-scanned image

When you process pre-scanned images that are produced on a system outside of FastDoc. You might need to know the FastDoc file name and the file name that is given to the image by the originating system.

Both the FastDoc file name and the original file name for each pre-scanned image are stored in the Scan.xml, Recognize.xml, Verify.xml and Export.xml files. These files are found in each batch folder. If there are multiple images in a batch, the names of all the images in the batch are found in these XML files.

- The original file name for each pre-scanned image is stored in the IMAGEORIGIN element.
- The FastDoc file name for each image is stored in the IMAGEFILE element.

Parent topic: [FastDoc maintenance](#)

Maintaining fingerprints by using the Fingerprint Maintenance Tool

Use the Fingerprint Maintenance Tool (FMT) to manage your fingerprints and synchronize information between the Fingerprint database, the Document Hierarchy, and Fingerprint XML data files.

About this task

Important: Before you start the Fingerprint Maintenance Tool, confirm that all Datacap clients are stopped.

If you attempt to start the Fingerprint Maintenance Tool and the DCO file is locked by another process, Datacap displays an error message. For example, when a developer is modifying the document hierarchy in Datacap Studio, the DCO file is locked and you cannot use the Fingerprint Maintenance Tool.

- [Starting the Fingerprint Maintenance Tool](#)
Start the Fingerprint Maintenance Tool on the Datacap server where the application files are located.
- [Locating and deleting partial fingerprints](#)
Partial fingerprints that are associated with documents that are not processed accumulate over time and must be deleted.
- [Deleting fingerprints](#)
You can use the Fingerprint Maintenance Tool to delete unwanted fingerprints from the system. You must know the Fingerprint ID.
- [Deleting fingerprints that are associated with a document type](#)
In FastDoc, fingerprints are saved by host name, where the host name is the same as the document type name. You must know the document type name before you run this procedure.
- [Exporting selected fingerprints](#)
When you want to move fingerprints from one application environment to another, such as from Test to Production, you must first export the fingerprints from the source Fingerprint database. You then import, or add, those fingerprints to the target Fingerprint database.
- [Adding selected fingerprints](#)
To move fingerprints from one application environment to another, such as from Test to Production, after you export the fingerprints from the source Fingerprint database. You then import, or add, those fingerprints to the target Fingerprint database.
- [Troubleshooting the Fingerprint Maintenance Tool](#)
This section contains information on resources that might help you to troubleshoot the Fingerprint Maintenance Tool.

Parent topic: [Administering](#)

Related reference:

[Fingerprint Maintenance Tool reference](#)

Starting the Fingerprint Maintenance Tool

Start the Fingerprint Maintenance Tool on the Datacap server where the application files are located.

About this task

The first time the Fingerprint Maintenance Tool is run, a message is displayed indicating a backup directory is created.

Procedure

To start the Fingerprint Maintenance Tool:

1. Open Windows Explorer and navigate to the Datacap\Application\dco_Application folder. For example, go to Datacap\APT\dco_APT.
2. Double-click Fingerprint Maintenance Tool.exe. The Fingerprint Maintenance Tool window opens.

Parent topic: [Maintaining fingerprints by using the Fingerprint Maintenance Tool](#)

Locating and deleting partial fingerprints

Partial fingerprints that are associated with documents that are not processed accumulate over time and must be deleted.

Procedure

To locate and delete partial fingerprints:

1. Start the Fingerprint Maintenance Tool.
2. Click Find Problems to locate existing problems.
3. When entries with X are displayed, click Select All, then click Delete Selected. The partial fingerprints are deleted.
4. Close the Fingerprint Maintenance Tool.

Parent topic: [Maintaining fingerprints by using the Fingerprint Maintenance Tool](#)

Deleting fingerprints

You can use the Fingerprint Maintenance Tool to delete unwanted fingerprints from the system. You must know the Fingerprint ID.

Procedure

To delete a fingerprint:

1. Start the Fingerprint Maintenance Tool.
2. Click the Show Statistics button to display information about all existing fingerprints.
3. Select the ID of the fingerprint that you want to delete.

4. Click the Delete Selected button.

Parent topic: [Maintaining fingerprints by using the Fingerprint Maintenance Tool](#)

Deleting fingerprints that are associated with a document type

In FastDoc, fingerprints are saved by host name, where the host name is the same as the document type name. You must know the document type name before you run this procedure.

Procedure

To delete all fingerprints that are associated with a document type:

1. Start the Fingerprint Maintenance Tool for the application. For example, select Start > IBM Datacap Developer Tools > Datacap Accounts Payable Fingerprint Maintenance Tool.
2. Click Show Statistics to display information about all existing fingerprints.
3. Click the Host Name column heading to sort the list by host name.
4. Highlight the records where host name is equal to the document type.
5. Click Delete Selected.

Parent topic: [Maintaining fingerprints by using the Fingerprint Maintenance Tool](#)

Exporting selected fingerprints

When you want to move fingerprints from one application environment to another, such as from Test to Production, you must first export the fingerprints from the source Fingerprint database. You then import, or add, those fingerprints to the target Fingerprint database.

Procedure

1. Start the Fingerprint Maintenance Tool that is associated with the source application environment.
2. Click the Show Statistics button to display information about the existing fingerprints.
3. Highlight the fingerprints that you want to export.
4. Click the Export Selected button. The Enter Identification Prefix window opens.
5. Enter 2 alphabetic characters. These characters become the first 2 characters of the exported fingerprint-related files. Then, click OK. The window closes and the exported files are placed in the application's \fingerprint\Fingerprint Export folder.
6. For instructions on adding these exported fingerprints to the target application environment, see [Adding selected fingerprints](#).

Parent topic: [Maintaining fingerprints by using the Fingerprint Maintenance Tool](#)

Adding selected fingerprints

To move fingerprints from one application environment to another, such as from Test to Production, after you export the fingerprints from the source Fingerprint database. You then import, or add, those fingerprints to the target Fingerprint database.

Procedure

1. Using Windows Explorer, copy the exported fingerprint-related files (TIFF, CCO, XML) from the source application's \fingerprint\Fingerprint Export folder to the target application environment's \fingerprint

- folder.
2. Start the Fingerprint Maintenance Tool that is associated with the target application environment.
 3. Click the Find FPXML Files button to locate fingerprints that are not in the target application environment's Fingerprint database.
 4. When one or more entries are displayed, click the Select All button, then click the Add Selected button. The fingerprints are added to the target application environment's Fingerprint database.
 5. Close the Fingerprint Maintenance Tool.

Parent topic: [Maintaining fingerprints by using the Fingerprint Maintenance Tool](#)

Troubleshooting the Fingerprint Maintenance Tool

This section contains information on resources that might help you to troubleshoot the Fingerprint Maintenance Tool.

- [FMT Backup Directory](#)
This folder is found in the application's dco_ directory. The folder contains the FMT.Log and backup copies of the SetupDCO file that are created when you run the Fingerprint Maintenance Tool.
- [FMT.Log](#)
This log contains processing details that identify the changes that are made and attempted by the Fingerprint Maintenance Tool.

Parent topic: [Maintaining fingerprints by using the Fingerprint Maintenance Tool](#)

FMT Backup Directory

This folder is found in the application's dco_ directory. The folder contains the FMT.Log and backup copies of the SetupDCO file that are created when you run the Fingerprint Maintenance Tool.

Parent topic: [Troubleshooting the Fingerprint Maintenance Tool](#)

FMT.Log

This log contains processing details that identify the changes that are made and attempted by the Fingerprint Maintenance Tool.

Parent topic: [Troubleshooting the Fingerprint Maintenance Tool](#)

Tworzenie aplikacji

- [Pierwsze kroki w tworzeniu aplikacji](#)
Jeśli dopiero rozpoczynasz przygodę z tworzeniem aplikacji Datacap, dowiedz się, które składniki Datacap wykorzystasz do budowy i konfiguracji aplikacji do przechowywania tak, aby spełniały Twoje potrzeby biznesowe.
- [Tworzenie aplikacji z programem Datacap](#)
Niniejszy kurs pozwala zapoznać się z pojęciami i czynnościami, które ułatwią tworzenie aplikacji Datacap. Zapoznając się z kolejnymi tematami, stworzysz aplikację do przetwarzania dokumentów podróży.
- [Tworzenie niestandardowych raportów w Przeglądarce raportów](#)
Dzięki Report Viewer możesz generować wysoce szczegółowe raporty na podstawie surowych danych statystycznych. Za pomocą interfejsu WWW Report Viewer możesz tworzyć niestandardowe raporty

odczytujące dane z wybranych kolumn i wyświetlające je w postaci tabelarycznej. Możesz dostosować raporty do swoich potrzeb, dodając w stopkach kolumn pola sum lub średnich i dołączając wykresy.

- [Tworzenie aplikacji Maintenance Manager](#)

Aplikacje Maintenance Manager programu Datacap umożliwiają monitorowanie partii, ustawianie powiadomień o statusach i automatyczne usuwanie ukończonych partii.

- [Odwołanie do obiektowych interfejsów API](#)

Możesz wykorzystać obiektowe interfejsy API w programie Datacap do tworzenia lub modyfikowania środowisk wykonawczych partii i hierarchii dokumentów, a także otrzymywania lub modyfikowania poziomów rozpoznanej pewności, wartości pól, wartości tekstowych i typów obiektów.

- [Model interfejsu API w programie Datacap Navigator](#)

Informacje pokrewne:

[Podsumowania bibliotek działań](#)

Getting started with application development

If you are new to Datacap application development or you want to learn how to create an application from scratch, start here. Datacap enables you to build and configure capture applications with several tools and components. You can use the Datacap Application Wizard to build new applications based on templates and configure them using prebuilt actions and rules using either Datacap FastDoc or Datacap Studio. You can further customize your applications by developing your own actions, rulesets, and tasks.

Before you begin

In order to work on your application:

- Your administrator needs to provide you with an installed and configured Datacap system, as well as an account with the correct privileges for application development.
- Spend some time defining your business requirements and learning about how you will use your Datacap capture application to satisfy those requirements. See [Business Requirements and Application Architecture](#).
- You will need at least one batch of all of the documents you want to capture with your application. The application setup process requires these samples to configure how the data will be captured, processed, and exported. The documents must take the form of image files that you scanned yourself or obtained from the appropriate area of your business.

Procedure

The following tasks represent the steps necessary for creating a new application for document capture:

1. Start the administrative version of Datacap FastDoc. Click Start > Programs > IBM Datacap Developer Tools > Datacap Fastdoc (Admin). Select the Local for the connection type and click Login. The FastDoc program opens.
2. Click on the Application Wizard icon in the FastDoc program header. The icon can be found next to your user name. The wizard will guide you through the steps for creating your application using one of the following application templates:
 - Use the Forms application template for structured images where you know the types of data that you want to capture and where that data is on each image. For example, a 1040EZ tax form and the types of data on the form, such as name and address, are in the same location on every 1040EZ form. The Forms application template sets up a workflow that you can match against document fingerprints.
 - Use the Learning application template for structured or semi-structured forms where you will be receiving the documents from third parties (for example: invoices). Each time a form from a new

source is identified, a fingerprint (template) will be created. Some or all of the field data is in the same location on the forms from a single source. The Learning application template sets up a workflow where you can add rules for Datacap to learn the different document formats as they are encountered.

For images where the data is not found, the verifier is prompted to click the image and identify where the data is located. This Click N Key process populates the data into the data set so that the Learning application can automatically find the data the next time that type of image is encountered. In our hotel example, after the unstructured hotel bill is processed, the zones are saved to capture data directly. Then, each time an unstructured image with the same format is encountered, the data is captured directly in the same way that data is captured from structured images with Forms template applications.

3. After completing the wizard, log out of FastDoc and then log in again using the Datacap Server connection type. On the login page, you can now choose your new application in the Application field, and since you are connecting to the Datacap Server, specify the login credentials provided to you by your administrator.

Important: Never select FormTemplate or LearningTemplate as your application on the FastDoc login page. These applications are templates for your custom applications, and changes to them

4. You'll need to configure your new application, which includes setting up fields, image enhancement, field recognition, and validation. If your application was created based on the Forms template, you'll also set up fingerprints. You will continue to use FastDoc for these tasks.
 - o For applications created using the Forms template, see [Forms template configuration](#).
 - o For applications created using the Learning template, see [Learning template configuration](#).
5. You will need to customize the workflow of your application to your business needs. [The Datacap workflow](#) consists of several tasks and jobs, including page identification, character recognition, field validation, verification, and export. These tasks use rules and actions to process a batch of your documents.
 - a. First, [configure your export options](#). You can export the data you capture to several different repositories, including IBM® FileNet® Content Manager, IBM Content Manager, and IBM Box.
 - b. Set up your workflow tasks and batch profiles, and add and configure rulesets for your application. See [Application configuration on FastDoc](#).
 - c. Open your application in Datacap Studio to customize your workflow task rules and actions. Click Start > Programs > IBM Datacap Developer Tools > Datacap Studio. For more information on how to edit rules and actions, see [Datacap Studio](#).
6. [Process documents](#) through your application in FastDoc.

Related concepts:

[Datacap application development](#)

Related information:

[Video: How to Build an Application using FastDoc](#)

Datacap application development

This tutorial introduces you to the concepts and tasks that help you to develop your Datacap applications. Throughout the tutorial, you develop an application to process travel documents.

- [Business Requirements and Application Architecture](#)
The first step in developing any Datacap application is to define the business requirements.
- [Datacap Studio](#)
Datacap Studio is the Datacap application development environment that provides the tools that you need to develop and test your application.
- [Document hierarchy](#)
Document hierarchy defines the structure of the documents that you are processing and how Datacap

processes each element within the structure. Document hierarchy is also referred to as the Setup DCO.

- [The Datacap workflow](#)
During the data capture process, documents go through a workflow that consists of several tasks, including page identification, character recognition, field validation, verification, and export. Some tasks require operator intervention, while other tasks run automatically.
- [Document input](#)
Datacap works primarily with TIFF image files. So, the first activity in any Datacap workflow is to convert the documents to TIFF format and insert the documents into an input repository.
- [Page Identification](#)
Page identification is one of the first steps in any Datacap application. All incoming pages are initially assigned the default page type Other. Before Datacap can assemble those pages into documents and extract data from the pages, it must determine the correct type for each page.
- [Rule Execution](#)
Rule execution refers to how rules are associated with specific objects in the document hierarchy and how Datacap processes a batch of documents.
- [Document assembly](#)
Datacap identifies incoming pages and assigns the correct page type by using fingerprint matching or one of the other identification methods. The next step assembles the batch of individual pages into documents according to the rules that are defined within the document hierarchy.
- [Data recognition](#)
Data recognition is the stage during which you locate the fields that you want to capture and then convert the fields into character-based data.
- [Data Validation](#)
Data validation determines whether captured data complies with the data integrity rules that are defined in your business requirements.
- [Data verification](#)
During verification, Datacap displays pages to an operator for manual checking and possible correction.
- [Data export](#)
Datacap can export data to a text file, an XML file, a database, a Document Management system, or a custom business process. The default output format is a text file, but you can use some actions to export data to a database and an XML file.
- [Application Debugging](#)
Application debugging requires that you review two runtime log files, which are the Rulerunner Service (RRS) log and the task log. The RRS log provides detailed information about each action and is most helpful to application developers. The task log documents internal calls and is used mostly by IBM software support.
- [Handling line item grids](#)
The techniques that you implemented rely upon data at predictable locations on the page. When you receive an invoice, you do not know how many items the invoice might contain. There might be just one item, or there might be a hundred items, possibly spanning multiple pages. Datacap includes actions to handle line items grids. You define the region on the page that might contain line items and define the structure of one line item. Datacap can then scan the region and locate all of the individual line items.
- [Smart parameters](#)
Smart parameters are action arguments that get evaluated at run time.
- [Text matching](#)
You can add flexibility to your applications by using text matching to identify pages and locate data.
- [Pattern Matching](#)
You can use Datacap pattern matching to identify pages and adjust misaligned or distorted images.
- [Workflow automation, routing, and automatic fingerprint generation](#)
You can configure Rulerunner to monitor the job queue and run background tasks like PageID, Profiler, and Export automatically whenever batches are pending.
- [Datacap Web Client and remote scanning](#)
You can now update your application by using Datacap Web Client Administrator and run a batch

through the entire workflow by using a combination of web components and Rulerunner.

- [Filter batches by group in the Job Monitor \(Datacap Web Client\)](#)
In the Datacap Web Client, you can filter batches by groups in the Job Monitor based on your ADSI, LDAP, or LLDAP group authentication.
- [Fingerprint Management](#)
Fingerprints are used both for page identification and for specifying recognition zones. The following topics review basic fingerprint functions, provide more details about the fingerprint database, and examine an alternative method for storing zone position information with fingerprint XML (FPXML) files. Later, you can update the TravelDocs application to use FPXML.
- [Configuring content classification for XML layout block parsing](#)
Some XML configuration file changes might be needed for IBM® Content Classification to properly parse the text blocks sent to it by the RunDecisionPlanForBlocks action.
- [Application translation](#)
You can translate the text in Datacap applications that is displayed in Datacap clients: Datacap Desktop, FastDoc (Job Monitor only), and Datacap Navigator. The following text can be translated: workflow names, job names, task names, shortcuts, descriptions, field names, document types, page types, and validation error messages.

Related tasks:

[Getting started with application development](#)

Business Requirements and Application Architecture

The first step in developing any Datacap application is to define the business requirements.

The process of defining business requirements includes these steps.

- Identifying the types of documents the application processes
- Identifying the page types that are associated with each document type
- Deciding what data you want to capture from each page
- Specifying the business rules that determine whether the captured data is valid or not
- Determining how to manage documents that have problems, including invalid structures, unrecognizable pages, nonconforming data, or low-confidence character recognition
- Deciding how you want to export or release the data at the end of the workflow

The following topics show how to develop the business requirements for a Datacap application. They show the general Datacap application architecture so that you can begin mapping the business requirements to the application model.

- [Business requirements development](#)
Before you start implementation, you need to define the business requirements through collaboration with the various stakeholders. Defining the business requirement involves examining the documents that you want to process, determining which fields to capture, and deciding what to do with captured data.
- [General Datacap application architecture](#)
Datacap applications are designed to scan, process, and verify the data in your documents.
- [TravelDocs: Business requirements](#)
Before you develop the application, review the documents and pages that the application processes, identify the fields to capture, and determine the other business requirements.

Parent topic: [Datacap application development](#)

Business requirements development

Before you start implementation, you need to define the business requirements through collaboration with the various stakeholders. Defining the business requirement involves examining the documents that you want to process, determining which fields to capture, and deciding what to do with captured data.

Datacap applications vary in their scale and complexity. But they all seek to capture data from structured documents, which are also known as Forms. The documents can be printed pages or electronic images, but the data on the page must be first located and then interpreted with maximum accuracy.

If you are processing various document types, you must decide whether the documents are pre-sorted or processed as a mixed batch. If they are presorted, you can simplify implementation by processing each type independently, either with a separate application or a separate workflow for each type. However, if they are processed as mixed batches, you need a more sophisticated system of page identification and document assembly.

Although the goal is to create a fully automated system, there are inevitably points at which manual intervention is required. The business requirements must specify how to determine whether the information is accurate and what to do when there is a problem. After you defined the business requirements, you can design the application.

The tutorial does not provide a detailed procedure for determining business requirements. Instead, the tutorial presents the general Datacap application architecture and then examines the documents to process as you develop the TravelDocs application. This sample application is designed to demonstrate basic techniques that implement the main steps in the Datacap application workflow.

Parent topic: [Business Requirements and Application Architecture](#)

General Datacap application architecture

Datacap applications are designed to scan, process, and verify the data in your documents.

Although each Datacap application is different, most include seven basic steps.

Table 1. Flow chart of seven basic steps of an application, from page input to data export

Application step	Description
Page input	Scan a batch of hardcopy pages or import electronic documents into your application. The output from this stage is a <i>batch</i> of individual TIFF image files. Each page is initially assigned the page type <i>Other</i> .
Page identification	Perform image enhancement to improve the image quality. Then, determine each page type, automatically or by displaying it to an operator for manual identification if necessary. The goal is to identify the page type, but not a variant (for example, an airline ticket, but not a ticket from a specific airline).
Document assembly	Organize the individual page files into a document according to predefined document definitions (for example, a form might have two required pages and an optional attachment). Run document integrity confirmation to ensure that each document satisfies the rules for that document type.
Data recognition	On each page, locate the data fields for that page type (for example, an airline ticket contains a passenger name, a departure airport). Then, use a Datacap recognition engine to obtain the character data for each field. The recognition engine indicates the degree of confidence for each character.

Application step	Description
Data validation	Check the validity of specific fields. For example, you can check for valid dates, valid field formats, and correct totals. You can also complete searches to ensure that a state abbreviation is valid, or a purchase order number matches an item in a purchase order database.
Data verification	Display low-confidence data and fields that failed validation to an operator for verification, correction, and exception handling. When the operator submits the batch, the application runs the validation rules again to ensure that all data satisfies the validation criteria.
Data export	Export the data or document images to a text file, an XML file, a database, a Document Management system, or the next stage in a workflow.

Parent topic: [Business Requirements and Application Architecture](#)

TravelDocs: Business requirements

Before you develop the application, review the documents and pages that the application processes, identify the fields to capture, and determine the other business requirements.

Throughout the tutorial, you are developing an application to process travel documents. The tutorial demonstrates the general techniques for implementing each of the basic steps in the application workflow (document input, page identification, validation, export).

- [Document types and page types](#)
The documents that you use in TravelDocs are simplified versions of typical travel-related documents that might be submitted with an employee expense report.
- [Required document structure](#)
When you examined each travel document, you identified pages that are required and optional.
- [Fields for each page type](#)
When you examined each sample page, you identified the fields of interest.
- [Permissible field values](#)
You can specify field values for your business requirements.
- [Business validation rules](#)
First, you define the structure of each document type and the fields that you want to capture from each page. Then, you define how you want to validate the captured data to determine whether the data meets the business requirements.
- [Data export format](#)
The last stage in developing the business requirements for the TravelDocs application is to specify the format of the captured data for export.

Parent topic: [Business Requirements and Application Architecture](#)

Document types and page types

The documents that you use in TravelDocs are simplified versions of typical travel-related documents that might be submitted with an employee expense report.

These documents include car rental receipts, hotel receipts, and air tickets. The document types and page types are summarized in the following table.

Table 1. Document types and page types in TravelDocs

Document type	Page types
Car Rental	Rental Agreement Optional Insurance
Hotel	Room Receipt Meals Other Charges
Flight	Air Ticket

To consider each document type, you need to look at the sample images that are installed in `\datacap\traveldocs\images` folder.

Car Rental

The car rental documents have one required page and one optional page. Initially, the application supports documents from three car rental companies: Car Rental #1, Car Rental #2, and Car Rental #3.

The three sample rental agreement pages in the `\datacap\traveldocs\images` folder are Car1.tiff, Car3.tiff, and Car5.tiff. The fields include the data that you want to extract. These fields are common to all pages, although the position of each field is different for each page.

```
Vendor: Car Rental #1
Pickup Date: Mon, Oct 4, 2010
Pickup Location: New York (JFK)
Return Date: Fri, Oct 8, 2010
Return Location: New York (JFK)
Car Type: Full size
GPS  Child Seat  Fuel Service 
Total Cost: $582.77
```

```
Vendor: Car Rental #2
Pickup Date: Sun, Aug 1, 2010
Pickup Location: Los Angeles (LAX)
Return Date: Fri, Aug 6, 2010
Return Location: Los Angeles (LAX)
Car Type: Luxury
GPS  Child Seat  Fuel Service 
Total Cost: $503.39
```

```
Vendor: Car Rental #3
Pickup Date: Sun, Oct 24, 2010
Pickup Location: Chicago (ORD)
Return Date: Fri, Fri, Oct 29, 2010
Return Location: Chicago (ORD)
Car Type: Compact
GPS  Child Seat  Fuel Service 
Total Cost: $535.18
```

The three sample optional insurance pages in the `\datacap\traveldocs\images` folder are Car2.tif, Car4.tif, and Car6.tif. The fields, for which you want to extract the data, are shown in these examples

```
Vendor: Car Rental #1
CDW: 
PAI: 
```

PEP:
ELP:
Total Cost: \$104.95

Vendor: Car Rental #2
CDW:
PAI:
PEP:
ELP:
Total Cost: \$0.00

Vendor: Car Rental #3
CDW:
PAI:
PEP:
ELP:
Total Cost: \$137.94

As with the rental agreement pages, the fields are common to all pages, but the position of each field is different for each variant.

Hotel

The hotel documents have one required page and two optional pages. Initially, the application supports documents from three hotel chains: Hotel #1, Hotel #2, and Hotel #3.

The three sample room receipts in the \datacap\traveldocs\images folder are Hotel1.tif, Hotel2.tif, and Hotel3.tif. As with the car rental pages, these fields are common to all pages, although the positions of the fields are different for each page.

Vendor: Hotel #1
Arrival Date: Sept 24, 2010
Departure Date: Sept 26, 2010
Total Cost: \$215.33

Vendor: Hotel #2
Arrival Date: Oct 14, 2010
Departure Date: Oct 16, 2010
Total Cost: \$282.51

Vendor: Hotel #3
Arrival Date: Sun, Oct 24, 2010
Departure Date: Tues, Oct 26, 2010
Total Cost: \$256.83

The following samples are the optional hotel pages Hotel4.tif and Hotel5.tif.

Vendor: Hotel #3
Item
 Date: 10-24-10
 Description: Dinner
 Cost: \$48.81
Item
 Date: 10-25-10
 Description: Breakfast
 Cost: \$12.28
Item
 Date: 10-25-10
 Description: Dinner
 Cost: \$46.41
Item
 Date: 10-26-10
 Description: Breakfast

Cost: \$12.28
Total Cost: \$119.78

Vendor: Hotel #3

Item

Date: 10-24-10
Description: Internet
Cost: \$5.95

Item

Date: 10-25-10
Description: Laundry
Cost: \$14.00

Item

Date: 10-25-10
Description: Internet
Cost: \$5.95

Item

Date: 10-26-10
Description: Parking
Cost: \$52.35

Total Cost: \$78.25

Flight

Flight documents have one required page and no optional pages. Initially, the application supports documents from three airlines: Airline #1, Airline #2, and Airline #3.

The three sample air ticket pages in the \datacap\traveldocs\images folder are Flight1.tif, Flight2.tif, and Flight3.tif. As with the other pages, these fields are common to all pages, although the positions of the fields are different for each page.

Vendor: Airline #1
Outbound From: New York/Newark (EWR)
Outbound To: San Francisco (SFO)
Outbound Date: 24JUL10
Return From: San Francisco (SFO)
Return To: New York/Newark (EWR)
Return Date: 28JUL10
Airfare: 760.27
Taxes: 64.56
Total Cost: 824.83

Vendor: Airline #2
Outbound From: Chicago (ORD)
Outbound To: Atlanta (ATL)
Outbound Date: MON OCT 25, 2010
Return From: Atlanta (ATL)
Return To: Chicago (ORD)
Return Date: WED OCT 27, 2010
Airfare: \$385.27
Taxes: \$44.76
Total Cost: \$430.03

Vendor: Airline #3
Outbound From: ORD Chicago
Outbound To: BOS Boston
Outbound Date: OCT 26, 2010
Return From: BOS Boston
Return To: ORD Chicago
Return Date: OCT 29, 2010
Airfare: 233.00 USD
Taxes: 21.40 USD
Total Cost: 254.40 USD

Required document structure

When you examined each travel document, you identified pages that are required and optional.

For example, in car rental documents:

- The rental agreement page is required.
- The insurance page is optional.

For travel documents with multiple pages, there might be requirements for the number or order of pages of each document type. This table summarizes the structure of each travel document type.

Document Type	Page Type	Number	Required?	Order
Car Rental		Any number per batch	No	Any position within batch
	Rental Agreement	One per document	Yes	Must be first in document
	Optional Insurance	One per document	No	Must be second in document
Hotel		Any number per batch	No	Any position within batch
	Room_Receipt	One per document	Yes	Must be first in document
	Meals	Any number per document	No	Cannot be first in document
	Other_Charges	Any number per document	No	Cannot be first in document
Flight		Any number per batch	No	Any position within batch
	Air_Ticket	One per document	Yes	Must be first in document

This structural information is an important element of the design requirements that you use when you implement the application's document hierarchy. When you implement the document assembly stage of the workflow, you use this information to determine whether the pages in the batch meet the structural requirements.

The assumption for the sample application is that you are entering batches of mixed travel documents with multiple, consecutive pages that are in the correct order. For example, a batch might include any number of car rental documents, flight documents, and hotel documents. Also, the pages within each document are consecutive and in the correct order. If the batch meets the structural requirements, then the application assembles the documents automatically. However, if the batch contains orphan pages or pages that do not meet the rules for document integrity, then operator intervention is required.

In the following example, the batch does not contain any errors, and no operator intervention is required.

Page type	Page type	Page type	Page type	Page type	Page type	Page type	Page type	Page type
Rental Agreement	Optional Insurance	Air Ticket	Room Receipt	Room Receipt	Meals	Rental Agreement	Optional Insurance	Air Ticket

In this second example, the batch contains three errors and requires operator intervention.

Page type	Page type	Page type	Page type	Page type	Page type	Page type	Page type	Page type
Optional Insurance (1)	Room Receipt	Room Receipt	Air Ticket	Meals (2)	Rental Agreement	Optional Insurance (3)	Optional Insurance (3)	Air Ticket

1. Orphaned optional insurance page must follow a rental agreement page.
2. Orphaned meals page must follow a room receipt page.

3. Two optional insurance pages are not allowed in a Car Rental document.

Parent topic: [TravelDocs: Business requirements](#)

Fields for each page type

When you examined each sample page, you identified the fields of interest.

You noted that each variant of a page type includes all of these fields, but the position of each field is different for each variant. This list summarizes the fields that you need to capture for each page type.

Car Rental document:

- Rental Agreement page type fields:
 - Vendor
 - Pickup_Date
 - Pickup_Location
 - Return_Date
 - Return_Location
 - Car_Type
 - Options
 - Nav_System
 - Child_Seat
 - Fuel_Service
 - Total_Cost
- Optional Insurance page type fields:
 - Vendor
 - Collision Damage Waiver
 - CDW_Option
 - Personal Accident Insurance
 - PAI_Option
 - Personal Effects Protection
 - PEP_Option
 - Extended Liability Protection
 - ELP_Option
 - Total_Cost

Hotel document:

- Room Receipt page type fields:
 - Vendor
 - Arrival_Date
 - Departure_Date
 - Total_Cost
- Meals page type fields:
 - [Item]
 - Date
 - Description
 - Cost
 - Total_Cost
- Other Charges page type fields:
 - [Item]
 - Date
 - Category

- Cost
- Total_Cost

Flight document:

- Air Ticket page type fields:
 - Vendor
 - Outbound_From, Outbound_To, Outbound_Date
 - Return_From, Return_To, Return_Date
 - Airfare
 - Taxes
 - Total_Cost

The two car rental pages both include check box options. There is a requirement in Datacap that each check box option is the child of a parent container field. On the Rental Agreement page, the three options are each a child of the same parent field. On the Optional Insurance page, each option has its own parent. The implementation is a little different depending on which method is used. So, this tutorial uses one of each method to demonstrate both techniques when you complete the implementation. The choice is more an implementation decision than a business decision, although it does affect the format of the export data.

Second, the optional hotel pages include repeating line items, each with the same structure. You do not know in advance how many items might be on a page. Datacap includes functionality for handling line item grids that are introduced in the topic [Handling line item grids](#).

Parent topic: [TravelDocs: Business requirements](#)

Permissible field values

You can specify field values for your business requirements.

In addition to specifying the fields, the business requirements might specify permissible formats and values for each field.

Page	Field	Permissible values
Rental Agreement	Vendor	Any text
	Pickup_Date	Any valid date format
	Pickup_Location	Any text
	Return_Date	Any valid date format
	Return_Location	Any text
	Car_Type	Compact, Standard, Full size, SUV, or Other
	Options	Checkbox fields - selected or not selected
	Total_Cost	Any currency format (\$999.99, 999.99, and 999.99 USD are valid)

Parent topic: [TravelDocs: Business requirements](#)

Business validation rules

First, you define the structure of each document type and the fields that you want to capture from each page. Then, you define how you want to validate the captured data to determine whether the data meets the business requirements.

For simplicity purposes in the sample application, you validate some of the fields only. You selected these fields specifically to demonstrate a few generic but commonly used techniques when you implement the data validation stage of the application workflow.

Table 1. Validation rules for sample application fields

Page	Field	Validation rule
Rental Agreement	Total Cost	Is the field value in a valid currency format? Specifically, is the field numeric with a two-digit decimal portion?
Optional Insurance	Total Cost	Is the field value in a valid currency format? Specifically, is the field numeric with a two-digit decimal portion?
Room Receipt	Total Cost	Is the field value in a valid currency format? Specifically, is the field numeric with a two-digit decimal portion?
Meals	Total Cost	Is the field value in a valid currency format? Specifically, is the field numeric with a two-digit decimal portion?
Other Charges	Total Cost	Is the field value in a valid currency format? Specifically, is the field numeric with a two-digit decimal portion?
Air Ticket	Air Fare Taxes Total Cost	Is the field value in a valid currency format? Specifically, is the field numeric with a two-digit decimal portion?
Rental Agreement	Car Type	Is the field value one of the following values: Compact, Standard, Full size, SUV, or Other?
Air Ticket	Air Fare Taxes Total Cost	Does the value of the Air Fare field plus the value of the Taxes field equal the value of the Total Cost field?

A validation failure does not necessarily mean that the original page contains invalid data. It might mean that the recognition engine failed to recognize one or more characters correctly. Whatever the reason for the error, the application developer can set the page status to ensure that the page is displayed to an operator for verification.

Parent topic: [TravelDocs: Business requirements](#)

Data export format

The last stage in developing the business requirements for the TravelDocs application is to specify the format of the captured data for export.

Datacap can export data to a text file, an XML file, a database, a Document Management system, or the input stage of another business application.

This example use case, which exports data only and does not export images, is not typical, and is used for simplicity. Almost all Datacap applications export images and documents together with captured data. In most

Document Management systems, the captured data is stored in metadata or index fields that are associated with each document.

For TravelDocs, you specify that data is to be exported to a Microsoft Access database and also saved in XML format. To simplify the implementation, you export only the rental agreement page data initially:

- For the database export, the application must export the data from each rental agreement page as a single record.
- For the XML export, all rental agreement pages in the same batch are written to a single XML file.

```
<?xml version='1.0' ?>
<Rental_Agreements>
  <TM000001>
    <Pickup_Date>Tues, Dec 7, 2010</Pickup_Date>
    <Pickup_Location>Boston (BOS)</Pickup_Location>
    <Return_Date>Fri, Dec 10, 2010</Return_Date>
    <Return_Location>Boston (BOS)</Return_Location>
    <Car_Type>Compact</Car_Type>
    <Options>Fuel Service</Options>
    <Total_Cost>$345.70</Total_Cost>
  </TM000001>
  <TM000003>
    <Pickup_Date>Mon, Dec 6, 2010</Pickup_Date>
    <Pickup_Location>San Francisco (SFO)</Pickup_Location>
    <Return_Date>Fri, Dec 10, 2010</Return_Date>
    <Return_Location>San Francisco (SFO)</Return_Location>
    <Car_Type>SUV</Car_Type>
    <Options>Child Seat</Options>
    <Total_Cost>$489.31</Total_Cost>
  </TM000003>
  <TM000004>
    <Pickup_Date>Mon, Dec 13, 2010</Pickup_Date>
    <Pickup_Location>Newark (EWR)</Pickup_Location>
    <Return_Date>Thur, Dec 16, 2010</Return_Date>
    <Return_Location>Newark (EWR)</Return_Location>
    <Car_Type>Other</Car_Type>
    <Options>Navigation System Child Seat Fuel Service</Options>
    <Total_Cost>$387.40</Total_Cost>
  </TM000004>
</Rental_Agreements>
```

In future tasks, you can export some of the line item grid data too.

Parent topic: [TravelDocs: Business requirements](#)

Datacap Studio

Datacap Studio is the Datacap application development environment that provides the tools that you need to develop and test your application.

Datacap Studio contains three main tabs: Rulemanager, Zones, and Test. In addition, Datacap Studio consists of an application wizard that you use to generate an application framework, which includes the supporting folder structure and control files.

- [Quick tour of the user interface](#)
Before you develop the TravelDocs application, you can review the Datacap Studio interface by opening one of the sample applications that is installed with Datacap.
- [TravelDocs: Start the TravelDocs application](#)
To start the application, you first need to create the application framework with the application wizard, and then connect to the application through Datacap Studio.

Parent topic: [Datacap application development](#)

Quick tour of the user interface

Before you develop the TravelDocs application, you can review the Datacap Studio interface by opening one of the sample applications that is installed with Datacap.

You use Datacap Studio extensively to develop the sample TravelDocs application.

- [Starting Datacap Server](#)
Depending on how your system is configured, the Datacap Server can start automatically or you might need to start it manually.
- [Opening a sample Datacap application](#)
After you confirm that the Datacap Server is running, you can start Datacap Studio and open any of the sample applications.
- [Panel organization within Datacap Studio](#)
Datacap Studio contains three main tabs, including the Rulemanager tab, the Zones tab, and the Test tab.
- [The Rulemanager tab](#)
The Rulemanager tab contains five panels where you define document structures, rulesets, rules, functions, and task profiles.
- [The Zones tab](#)
The Zones tab contains four panels where you can add fingerprints and view properties of selected objects.
- [The Test tab](#)
The Test tab contains eight panels in which you can view information and properties of batches, jobs, documents, and rulesets.

Parent topic: [Datacap Studio](#)

Starting Datacap Server

Depending on how your system is configured, the Datacap Server can start automatically or you might need to start it manually.

About this task

Datacap applications use required services (such as authentication, batch creation, and assignment) that are provided by the Datacap Server, which runs in the background as a Windows service. If the server is not running, you cannot log on to Datacap.

To start the Datacap Server:

Procedure

1. In the Start menu click IBM Datacap Services > Datacap Server Manager.
2. On the Service tab in Datacap Server Manager, verify that the status is *Running*.
3. If the status is not *Running*, click the Start button.
4. Confirm that the status is *Running* and then click Close. The server is now running in the background.

Parent topic: [Quick tour of the user interface](#)

Opening a sample Datacap application

After you confirm that the Datacap Server is running, you can start Datacap Studio and open any of the sample applications.

About this task

To open one of the sample applications in Datacap Studio:

Procedure

1. In the Start menu click IBM Datacap Developer Tools>Datacap Studio.
2. In the Select Application window, select one of the existing sample applications, and click Next. For example, one of the existing sample applications is TravelDocs.
3. In the Datacap Login window, ensure that the NT authentication check box is not selected.
4. Enter these values for the fields as shown.
 - o User ID:admin
 - o Password:admin
 - o Station ID:1
5. Click Finish.

Parent topic: [Quick tour of the user interface](#)

Panel organization within Datacap Studio

Datacap Studio contains three main tabs, including the Rulemanager tab, the Zones tab, and the Test tab.

Tab	Description
Rulemanager	This tab is the primary application development area.
Zones	This tab is where you create page fingerprints and configure recognition zones.
Test	This tab provides integrated execution and debugging tools for testing your application.

Each main tab contains more tabs and panes.

You can customize the workplace by reorganizing the panes, removing panes, and adding panes.

To move a pane:

1. Use the mouse to drag the pane's tab from its current location. You see a set of insertion points that are located around the window. You can move the pane to the left, right, top, or bottom of another pane, or to the left, right, top, or bottom of the window. In the center, you can combine tabs. As you move the pointer over an insertion point, you see a shaded area that indicates the corresponding location.
2. Drop the pane on an insertion point to move the pane.

To remove a pane, right-click the pane's tab and choose Close.

To add a pane, right-click any tab and choose Show tabs. Then, choose from the available panes. After you add a pane, you can move the new pane.

Parent topic: [Quick tour of the user interface](#)

The Rulemanager tab

The Rulemanager tab contains five panels where you define document structures, rulesets, rules, functions, and task profiles.

The Rulemanager tab includes these panels:

Table 1. Rulemanager tab panels

Panel	Description
Document hierarchy	Defines the structure of the documents you are processing and how each element within the structure is processed (see Document hierarchy).
Rulesets	Defines the rules, functions, and actions that make up each ruleset (see Rulesets, rules, and actions).
Task profiles	Defines the rulesets that are run by each task profile (see Task profiles and rulesets).
Actions library	Provides access to the complete library of pre-built actions and, in some cases, custom developed actions. To get help on an action, select the action and click the Information icon.
Properties	Displays the properties for the selected document hierarchy or ruleset object. If the corresponding pane is locked for editing, you can also modify existing properties, including specifying action parameters.

Parent topic: [Quick tour of the user interface](#)

The Zones tab

The Zones tab contains four panels where you can add fingerprints and view properties of selected objects.

The Zones tab includes these panels:

Table 1. Zones tab panels

Panel	Description
Fingerprints	Displays the application's fingerprint library from which you can add fingerprints for new page types (see Fingerprint matching).
Document hierarchy	Defines the structure of the documents that you are processing and how each element within the structure is processed. (See Document hierarchy .)

Panel	Description
Properties	Displays the properties for the selected document hierarchy object. If the document hierarchy is locked for editing, you can also modify existing properties. In the Properties panel, you can specify recognition options for the selected object. Datacap supports multiple recognition engines. The Properties panel displays the ICR/C, BAR/P, and OCR/S tabs by default. You can access other tabs by right-clicking within the Properties panel and selecting Show tabs.
Image View	Displays the selected fingerprint image and any recognition zones. Also, you can draw new recognition zones in the Image View panel (see Identifying recognition zones by using fingerprints). If you created the fingerprints by using full page recognition, you can view the recognition results in the Text tab.

Parent topic: [Quick tour of the user interface](#)

The Test tab

The Test tab contains eight panels in which you can view information and properties of batches, jobs, documents, and rulesets.

The Test tab includes these panels:

Table 1. Test tab panels

Panel	Description
Workflow	Displays the job types and tasks that are defined in the Administrator tab. Also, you can run a batch through the workflow in the Workflow panel.
Runtime batch hierarchy	When a batch is running, this panel displays the runtime batch hierarchy, including any data values. If you select a page object, the page is displayed in the Image panel.
Document hierarchy	Displays the structure of the documents that you are processing, and shows how each element within the structure is processed.
Rulesets	Displays the rules, functions, and actions that make up each ruleset. As you step through the workflow, you can see the current execution point.
Image/Text	Displays the selected page in the runtime batch hierarchy.
Batch data	Displays batch level information for the batch that is running.
Properties	Displays the properties for the selected document hierarchy or ruleset object (read only).
Breakpoints/Runtime state/Call stack	A breakpoint stops processing at a predetermined ruleset, rule, or action. For more information, see Using breakpoints .

Parent topic: [Quick tour of the user interface](#)

TravelDocs: Start the TravelDocs application

To start the application, you first need to create the application framework with the application wizard, and then connect to the application through Datacap Studio.

- [The application framework](#)
You can create an application, copy an application, or convert an application format from a previous version by using the Datacap application wizard in Datacap Studio.
- [Connecting to the application](#)
When you create a new application by using the application wizard, the application is added to the list of applications in the Datacap Application Manager.

Parent topic: [Datacap Studio](#)

The application framework

You can create an application, copy an application, or convert an application format from a previous version by using the Datacap application wizard in Datacap Studio.

You can create or copy an application, including a CMIS-based application, when you run the application wizard in Datacap Studio. You can also convert an 8.0.1 application to a 9.0 format. You do not need to convert an 8.1 application to a 9.0 format.

Select the Forms or Learning application template.

- Select the Forms application template for structured images. When you know the types of data that you want to capture and where that data is on each image, select the Forms application template. For example, a 1040EZ tax form and the types of data on the form, such as name and address, are in the same location on every 1040EZ form. The Forms application template sets up a workflow that you can match against your fingerprints.
- Select the Learning application template for unstructured images. When you know the types of data that you want to capture but you do not know where that data is contained in the image because the location of the data is different on each image, select the Learning application template. For example, if you want to capture the date, amount, and tax for expenses from different hotels, the receipt images from each hotel are unique. The location of the data you want to capture differs for each hotel receipt image so the data cannot be identified with Datacap fingerprints. The Learning application template sets up a workflow where you can add rules, such as Locate rules, for Datacap to learn the different hotel receipt formats as they are encountered.
 - For images where the data is not found, the verifier is prompted to click the image and identify where the data is located. This Click N Key process populates the data into the data set so that the Learning application can automatically find the data the next time that type of image is encountered. After the unstructured hotel bill is processed, the zones are saved to capture data directly. Then, each time an unstructured image with the same format is encountered, the data is captured directly in the same way that data is captured from structured images with Forms applications.

Parent topic: [TravelDocs: Start the TravelDocs application](#)

Connecting to the application

When you create a new application by using the application wizard, the application is added to the list of applications in the Datacap Application Manager.

About this task

Before you can work with the application in Datacap Studio, you must connect to the application.

Procedure

To connect to the application from Datacap Studio:

1. In Datacap Studio, click the Connection Wizard button.
2. Select the MyTravelDocs application and click Next.
3. Log in by using User ID: admin, Password: admin, and Station: 1.
4. Click Finish. Datacap Studio displays the new project.

Parent topic: [TravelDocs: Start the TravelDocs application](#)

Document hierarchy

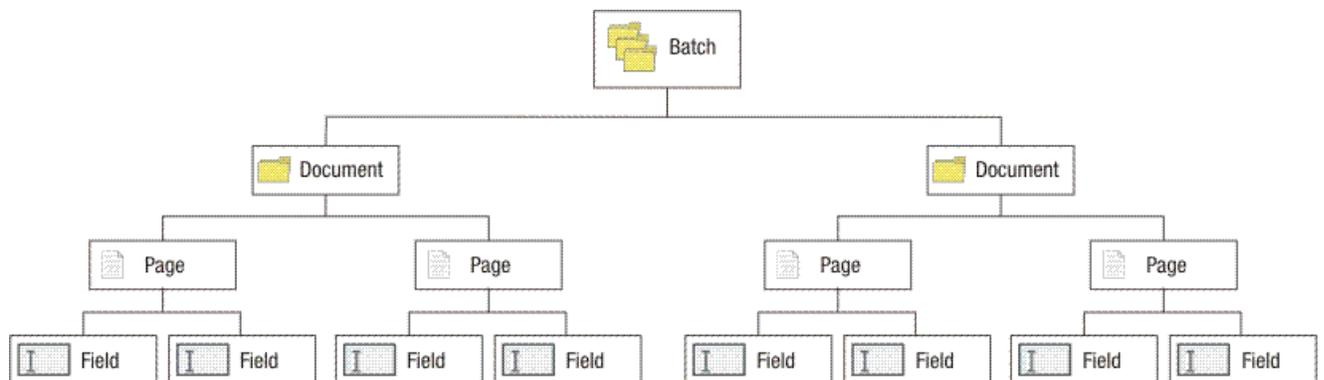
Document hierarchy defines the structure of the documents that you are processing and how Datacap processes each element within the structure. Document hierarchy is also referred to as the Setup DCO.

- [Document structure](#)
The document hierarchy describes the structure of the documents that your application is designed to process. The levels within the hierarchy are batch, document, page, and field.
- [Identification of page types from documents](#)
There are several techniques to identify individual page types, but the most common technique is called *fingerprint matching*.
- [Relation of the document hierarchy to the runtime batch hierarchy](#)
The document hierarchy describes the general structure of the documents that your application supports in terms of document types, page types, and fields. By contrast, a runtime batch describes specific documents that contain specific pages and specific data.
- [Page type versions](#)
Although individual pages within a runtime batch might be of the same type, the pages might look different.
- [TravelDocs: Create the document hierarchy](#)
The document hierarchy enables the Datacap application to convert a collection of unstructured images into a structured runtime batch hierarchy that contains the relevant business data.

Parent topic: [Datacap application development](#)

Document structure

The document hierarchy describes the structure of the documents that your application is designed to process. The levels within the hierarchy are batch, document, page, and field.



At the top of the document hierarchy is the batch, which refers to all pages of all document types. Beneath the batch level, the document hierarchy defines:

The document types your application can process

An application can process only one document type, or multiple document types. For example, the TravelDocs sample application can process car rental documents, hotel expense documents, and flight documents.

The page types within each document type

Each document can contain only one page type or multiple page types. For example, the TravelDocs car rental document includes the rental agreement page and the optional insurance page, while the flight document has only an air ticket page.

The number and order of pages within each document type

Pages can be required or optional. For example, a car rental document has two pages at most. The rental agreement page is required and must come first; and the insurance coverage page is optional.

The data fields within each page type

Data fields can be required or optional. For example, the hotel document's Other Charges page has fields for expense category, number of items, unit cost, and total cost. Nested fields are used for special purposes and creation of custom nested fields is not supported.

Parent topic: [Document hierarchy](#)

Identification of page types from documents

There are several techniques to identify individual page types, but the most common technique is called *fingerprint matching*.

In a typical Datacap application, documents start as a batch of unidentified image files with one image per page. A single batch might contain a mix of document types, and each document might contain a number of different page types. There is nothing within the page image that identifies the page type or any of the data on the page. In other words, the page images do not contain any structured content.

Before Datacap can begin to extract data, it must identify the individual page types. Datacap then maps pages to documents, and fields to pages, by using the information in the document hierarchy. After Datacap identifies the fields and their locations within each page, it extracts and stores the data in a structured format. The structured format is known as the *runtime batch hierarchy*.

Parent topic: [Document hierarchy](#)

Relation of the document hierarchy to the runtime batch hierarchy

The document hierarchy describes the general structure of the documents that your application supports in terms of document types, page types, and fields. By contrast, a runtime batch describes specific documents that contain specific pages and specific data.

The document hierarchy and the runtime batch can be described in object-oriented terms:

- The document hierarchy defines the document, page, and field classes.
- The runtime batch describes a set of objects that is built from those classes. Each object has a set of variables that is derived from the parent class, and each variable has a value.

While the document hierarchy describes a single, generalized version of each document and page type, a runtime batch can have any number of documents and pages.

In the TravelDocs application, the document hierarchy defines the three document types that include Car_Rental, Hotel, and Flight. The runtime batch might include two car rental documents, two hotel

documents, and two flight documents. Each runtime document has one or more pages. Each page has the number of fields that are defined in the document hierarchy for that page type.

Parent topic: [Document hierarchy](#)

Page type versions

Although individual pages within a runtime batch might be of the same type, the pages might look different.

The TravelDocs runtime batch hierarchy includes two car rental documents, two hotel documents, and two flight documents. The car rental documents might be from different car rental companies. The hotel documents might be from different hotel chains, and the flight documents might be from different airlines.

For example, the TravelDocs runtime batch has two pages of type Rental_Agreement. (Review the files Car1.tif and Car3.tif in \Datacap\TravelDocs\images.) Structurally, the pages contain the same data.

Occasionally, the location of data is not in the same position on different pages. To identify the location of data, you can create a fingerprint for each variant and store the field location for each variant in the document hierarchy.

Parent topic: [Document hierarchy](#)

TravelDocs: Create the document hierarchy

The document hierarchy enables the Datacap application to convert a collection of unstructured images into a structured runtime batch hierarchy that contains the relevant business data.

The goal of this part of the tutorial is to create generalized definitions (classes) for the document types, page types, and fields that the application supports.

- [Default document hierarchy](#)
The Datacap Studio application wizard creates a default document hierarchy that you can use as a starting point.
- [Creating document types](#)
The business requirements specification for the TravelDocs application defines three document types that include Car rental, Hotel, and Flight.
- [Creating page types](#)
You need to create at least one page type for each of the three different document types that are contained in the TravelDocs document hierarchy.
- [Specifying the structure of documents and pages within the batch](#)
In addition to creating page types for each document type, you need to configure rules and variables for the pages and documents.
- [Creating data fields](#)
Each page type requires multiple field definitions.
- [Specifying the structure of fields on each page](#)
The business requirements for the TravelDocs application specify that there must be only one instance of each field on each page. The order of the fields within the page is not important, but each field needs to be configured with Max=1, Min=1, and Order=0. The default values for fields are Max=0, Min=0, and Order=0.
- [Sharing field definitions across the document hierarchy](#)
The document information, page information, and field information are stored in a file referred to as the document hierarchy or the setup DCO.

Parent topic: [Document hierarchy](#)

Default document hierarchy

The Datacap Studio application wizard creates a default document hierarchy that you can use as a starting point.

The default hierarchy includes these objects:

- A batch node that has the same name as the application
- A page type Other, which is the default type that Datacap assigns to all pages before page identification
- A default document type called Document
- A default page type called Page
- One default field that is called Field and is associated with the page type Page

In the document hierarchy, the Open and Close nodes define the rules that are assigned to each element within the hierarchy. For example, the Open node beneath the page type Other defines the rules and actions that Datacap starts when it begins processing a page of type Other.

Parent topic: [TravelDocs: Create the document hierarchy](#)

Creating document types

The business requirements specification for the TravelDocs application defines three document types that include Car rental, Hotel, and Flight.

About this task

You begin by adding these document types to the hierarchy.

Procedure

To add these document types to the hierarchy:

1. In the Document Hierarchy pane, click Lock DCO for editing to lock the document hierarchy for editing.
Tip: The terms *DCO* and *document hierarchy* are used interchangeably.
2. Expand the tree so that you can see the default document and page types.
3. Select and single-click the Document node to edit the name.
4. Change the name from Document to Car_Rental and press Enter.
Important: You cannot include spaces in any of the document hierarchy node names.
5. Right-click the TravelDocs batch node and choose Add multiple >> Documents. Then, type 2 in the box and press Enter.
6. Rename the new documents from Document1 and Document2 to Flight and Hotel.
7. Click Save.

Parent topic: [TravelDocs: Create the document hierarchy](#)

Creating page types

You need to create at least one page type for each of the three different document types that are contained in the TravelDocs document hierarchy.

About this task

The business requirements specification defines the following page types for each document type:

Table 1. Page types for each document type

Document types	Page types
Car_Rental	Rental_Agreement ✓ Optional_Insurance ✓
Hotel	Room_Receipt ✓ Meals x Other_Charges x
Flight	Air_Ticket ✓

To simplify the application, you can skip the Meals and Other_Charges pages.

Procedure

To create new page types:

1. Confirm that the document hierarchy is still locked for editing.
2. Beneath the Car_Rental document node, select the default Page node and change the name from Page to Rental_Agreement.
3. Right-click on the Car_Rental document node and choose Add > Page. Then, change the name of the page from Page1 to Optional_Insurance.
4. Right-click the Flight document node and choose Add > Page. Then, expand the Flight node and change the name of the page from Page1 to Air_Ticket.
5. Right-click the Hotel document node and choose Add > Page. Then, expand the Hotel node and change the name of the page from Page1 to Room_Receipt.
6. Click Save.

Parent topic: [TravelDocs: Create the document hierarchy](#)

Specifying the structure of documents and pages within the batch

In addition to creating page types for each document type, you need to configure rules and variables for the pages and documents.

About this task

Car Rental document

- Rental Agreement page
- Optional Insurance page

Flight document

- Air_Ticket page

Hotel document

- Room_Receipt page

The business requirements specify the following rules for the structure of each document type:

Table 1. Structural rules for each document type

Document type	Number	Required?	Order
Car Rental	Any number per batch	No	Any position within batch
Rental Agreement	One per document	Yes	Must be first in document
Optional Insurance	One per document	No	Cannot be first in document
Flight	Any number per batch	No	Any position within batch
Air_Ticket	One per document	Yes	Must be first in document
Hotel	Any number per batch	No	Any position within batch
Room_Receipt	One per document	Yes	Must be first in document

Within the document hierarchy, the following variables define the structure of the batch. By using these variables, you can define the structure of the batch.

Table 2. Variables defining the structure of the batch

Variable	Description
<i>Max</i>	Maximum number of objects of this type for each parent object. 0 means no maximum; 1 means Datacap creates a new document each time it encounters a page of this type, and so forth.
<i>Min</i>	Minimum number of objects of this type for each parent object. 0 means no minimum; 1 means there must be at least one, and so forth.
<i>Order</i>	Position of this object relative to other child objects of the same parent. 0 means any position.

Table 3. Batch structure variable values for each document type

Document type	<i>Max</i>	<i>Min</i>	<i>Order</i>
Car Rental	0	0	0
Rental Agreement	1	1	1
Optional Insurance	1	0	2
Flight	0	0	0
Air_Ticket	1	1	1
Hotel	0	0	0
Room_Receipt	1	1	1

To specify the structure of documents and pages within the batch:

Procedure

1. Confirm that the document hierarchy is still locked for editing.
2. Right-click the Car_Rental document node and choose Manage variables.
3. Set the Max, Min, and Order values (The Car_Rental document is 0, 0, 0.), and click Done.

4. Right-click the Rental_Agreement page node and choose Manage variables.
5. Enter the Max, Min, and Order values. (The Rental_Agreement page is 1, 1, 1.), and click Done.
6. Repeat for each of the remaining document and page types.
7. Click Save.

Parent topic: [TravelDocs: Create the document hierarchy](#)

Creating data fields

Each page type requires multiple field definitions.

About this task

The business requirements specification defines the following fields for each page type:

Table 1. Fields for each page type

Rental_Agreement	Optional_Insurance	Air_Ticket	Room_Receipt
Vendor x	Vendor x	Vendor x	Vendor x
Pickup_Date ✓	CDW ✓	Outbound_From ✓	Arrival_Date ✓
Pickup_Location ✓	CDW_Option ✓	Outbound_To ✓	Departure_Date ✓
Return_Date ✓	PAI ✓	Outbound_Date ✓	Total_Cost ✓
Return_Location ✓	PAI_Option ✓	Return_From ✓	
Car_Type ✓	PEP ✓	Return_To ✓	
Options ✓	PEP_Option ✓	Return_Date ✓	
Nav_System ✓	ELP ✓	Airfare ✓	
Child_Seat ✓	ELP_Option ✓	Taxes ✓	
Fuel_Service ✓	Total_Cost x	Total_Cost ✓	
Total_Cost ✓			

To simplify the application slightly, you can skip the fields marked x.

Procedure

To create data fields:

1. Confirm that the document hierarchy is still locked for editing.
2. Expand the Rental_Agreement page, select the default Field node, and change the name from Field to Pickup_Date.
3. Right-click the Rental_Agreement page and choose Add multiple > Fields.
4. Type 6 in the box and press Enter.
5. Rename the new fields Pickup_Location, Return_Date, Return_Location, Car_Type, Options, and Total_Cost.
6. Right-click the Options field and choose Add multiple > Fields.
7. Type 3 in the box and press Enter.
8. Expand the Options and rename the new fields Nav_System, Child_Seat, and Fuel_Service.
9. Click Save.
10. Use the same procedure to add the fields to the Optional_Insurance page. The Optional_Insurance page has four fields, each of which has one subfield.

11. Click Save. The Rental_Agreement, Room_Receipt, and Air_Ticket pages all have a field that is called Total Cost. When you add this field to the Room_Receipt and Air_Ticket pages, Datacap Studio displays a message that prompts you to reference the existing object. Click Yes. You see the same message when you add the Return_Date field to the Air_Ticket page. Click Yes again. For an explanation, see [Sharing field definitions across the document hierarchy](#).
12. Repeat these steps for the Air_Ticket and Room_Receipt pages to add the fields marked ✓ in the table. The Air_Ticket page has nine fields and the Room_Receipt page has three fields.
13. Click Save after each page.
14. Click Save. The complete document hierarchy for TravelDocs includes three document types, each of which contain at least one page type and multiple fields.

Parent topic: [TravelDocs: Create the document hierarchy](#)

Specifying the structure of fields on each page

The business requirements for the TravelDocs application specify that there must be only one instance of each field on each page. The order of the fields within the page is not important, but each field needs to be configured with Max=1, Min=1, and Order=0. The default values for fields are Max=0, Min=0, and Order=0.

About this task

There is no requirement to specify the structure of fields on each page because most applications create only one field of each type. Also, structure verification is only applied at the document and page level.

This procedure is optional. Specifying the values for all of the fields in the TravelDocs application is not required for this tutorial because all of the sample pages comply.

Procedure

To specify the structure of fields within each page:

1. Right-click each field node and choose Manage variables.
2. Set Max=1, Min=1, and Order=0 and then click Done.
3. After you complete the previous steps for all of the fields in the document hierarchy, including the subfields, click Save.
4. Click Unlock DCO.

Parent topic: [TravelDocs: Create the document hierarchy](#)

Sharing field definitions across the document hierarchy

The document information, page information, and field information are stored in a file referred to as the document hierarchy or the setup DCO.

About this task

You specify the document, page, and field information in the document hierarchy panel. Datacap Studio saves this information in the C:\Datacap\application_name\dco_application_name\application_name.xml file.

The document hierarchy for the TravelDocs application is C:\Datacap\TravelDocs\dco_TravelDocs\TravelDocs.xml. This file defines the structure of the batch, and the structure of each document type, page type, and field within the batch. Although the batch structure is

hierarchical, the structure of the file is flat, and the name of each document, page, and field object must be unique.

Within the document hierarchy file, the object definition specifies the child objects that are referenced by each parent object. For example, the Room Receipt definition specifies that a room receipt page has three child fields:

```
<P type="Room_Receipt">
  <V n="rules"></V>
  <F type="Arrival_Date" pos="0" min="0" max="0"/>
  <F type="Departure_Date" pos="0" min="0" max="0"/>
  <F type="Total_Cost" pos="0" min="0" max="0"/>
</P>
```

This structure allows multiple parent objects to reference the same child object.

In the TravelDocs application, the Rental Agreement, Air Ticket, and Room Receipt pages all have a Total_Cost field. The first time that you add the Total_Cost field to a page, Datacap adds the field to the document hierarchy. Later, when you add the field to the other page types, Datacap displays a message dialog that prompts you to use the existing reference. If you select Yes, all rules and properties are inherited.

Parent topic: [TravelDocs: Create the document hierarchy](#)

The Datacap workflow

During the data capture process, documents go through a workflow that consists of several tasks, including page identification, character recognition, field validation, verification, and export. Some tasks require operator intervention, while other tasks run automatically.

These topics examine how the Datacap queuing mechanism moves batches of documents through the workflow and how tasks are implemented programmatically in terms of rulesets, rules, and actions.

- [Understanding the Datacap workflow](#)

A workflow contains jobs and tasks. Furthermore, tasks are associated with task profiles that contain rules and actions that are applied by the tasks while a job is processing a batch.

Parent topic: [Datacap application development](#)

Understanding the Datacap workflow

A workflow contains jobs and tasks. Furthermore, tasks are associated with task profiles that contain rules and actions that are applied by the tasks while a job is processing a batch.

- [Workflows, jobs, and tasks](#)

A workflow consists of a series of tasks, defines a way to process documents, and is associated with only one DCO.

- [Task profiles and rulesets](#)

Each task is linked to a task profile that includes one or more rulesets. The default rulesets generated by the Application Wizard are displayed in the Task Profiles pane on the Datacap Studio Rulemanager tab.

- [Rulesets, rules, and actions](#)

A ruleset consists of one or more rules. The rule itself is defined by the programmed functions and actions within it.

Parent topic: [The Datacap workflow](#)

Workflows, jobs, and tasks

A workflow consists of a series of tasks, defines a way to process documents, and is associated with only one DCO.

Although Datacap applications can include multiple workflows, this tutorial focuses on single workflow applications. The standard workflow generated by the Application Wizard includes three job types:

- **Main Job:** This is the standard workflow for processing documents that takes a batch of documents through each of the processing steps that are previously identified, such as input documents, identify pages, and so on.
- **Fixup Job:** This job is used only when there are document integrity problems and displays the batch to an operator for corrective action. For more information, see the [Document integrity problem management](#) topic.
- **Web Job:** This job is like the Main Job, but it defines the workflow for jobs that are initiated exclusively from the Datacap Web Client. It supports remote scanning and allows users to upload new batches to the server.

A job consists of one or more tasks. To process a batch of documents, you must run the batch through each task in the selected job. Some tasks (for example, Export) run without operator intervention, whereas others (for example, Verify) require an operator.

The tasks in the workflow are determined by the job type you select. You can see the tasks associated with each job type by looking in the Workflow pane on the Datacap Studio Test tab. The workflow for Main Job includes five tasks: VScan, PageID, Profiler, Verify, and Export. Each task is linked to a task profile.

Descriptions of each task are provided.

Table 1. Main Job task descriptions

Task Profile	Description
VScan	A virtual scanning profile that inserts pages into your application by copying images files from a specified location.
Upload	Used with remote scanning and virtual scanning through the Datacap Web Client interface, the Upload task is required for uploading images from remote scanning stations to the batch folder on the Datacap server.
PageID	Identifies the incoming pages by comparing them to known page types using fingerprint matching. Depending on the identification method used, this profile may perform full page OCR. It may also perform image cleanup.
Profiler	Organizes pages into documents, locates the fields defined for that page type, and performs OCR to recognize the field data (or obtains the data from the full page OCR results). Also runs validation rules to ensure that the data is valid.
Verify	Runs during the verification stage, when pages are displayed to an operator to ensure that recognition was accurate and to handle any validation errors.
Export	Exports the structured document data to an output file, a document management system, a database, or an external business process (can also include the original image).

In addition to the task profiles that run as part of the Main Job workflow, there are two other important task profiles the Application Wizard generates: FingerprintAdd and ImageFix.

Table 2. Additional task profiles

Task Profile	Description
Fingerprint Add	Generates the fingerprint files when you add new page types to the application from the Datacap Studio Zones tab.
ImageFix	Runs when you enhance a fingerprint image using the Image Processing window from the Zones tab.

Parent topic: [Understanding the Datacap workflow](#)

Task profiles and rulesets

Each task is linked to a task profile that includes one or more rulesets. The default rulesets generated by the Application Wizard are displayed in the Task Profiles pane on the Datacap Studio Rulemanager tab.

The default Main Job workflow uses all of these task profiles, in the order as shown.

- VScan
- PageID
- Profiler
- Verify
- Export
- FingerprintAdd
- ImageFix

The FingerprintAdd profile runs when you add a new fingerprint to the application from the Zones tab.

The Imagefix profile runs when you enhance a fingerprint image by using the Image Processing window from the Zones tab.

Each ruleset defines one or more rules that you can run on specific documents, pages, or fields, or on the entire batch. The task profile specifies only that certain rulesets are associated with that profile. Nothing runs until you actually associate a specific rule with specific document, page, or field, or with the batch, as described in [Rule Execution](#).

Within each task profile, rulesets run in the order, although a ruleset will not do anything if the rules in it are not associated with any objects in the document hierarchy.

Attention: The order of the rulesets within the task profile is important, because it defines the order in which Datacap runs rules. For example, you cannot check the integrity of a document before you create the document. So, the CreateDocs ruleset must come before the Document Integrity ruleset.

Multiple task profiles can reference the same ruleset. For example, the Profiler and Verify profiles both reference the Validate ruleset because you typically run validation rules after data recognition, and run the same rules again after verification by the operator.

Parent topic: [Understanding the Datacap workflow](#)

Rulesets, rules, and actions

A ruleset consists of one or more rules. The rule itself is defined by the programmed functions and actions within it.

The default PageID ruleset has two rules, which are PageID and Set Fingerprint parameters. You can see the rules that are associated with each ruleset in the Rulesets panel on the Datacap Studio Rulemanager tab.

Rules are assigned to process specific objects in the document hierarchy (for example, to analyze and identify each page).

The default PageID rule consists of one function and two actions. The PageID function first launches the AnalyzeImage action. If AnalyzeImage is successful (returns True), the function launches the FindFingerprint. If AnalyzeImage fails (returns False), the function fails and Datacap launches the next function within the rule. In this case, there is not another function, but you could add an exception handling function to handle the error. See [Rule Execution](#).

When you login to an application, Datacap Studio searches for and imports the rulesets from DLLs that are not in the collection.xml file yet. It searches the Rules folder for the application first, then it searches the central RRS folder. During this operation, the Update Status bar indicates that the current file is being processed with a rotating spinner. When the spinner is gone, the operation is finished.

Parent topic: [Understanding the Datacap workflow](#)

Document input

Datacap works primarily with TIFF image files. So, the first activity in any Datacap workflow is to convert the documents to TIFF format and insert the documents into an input repository.

Documents can be hardcopy or electronic. If the documents are hardcopy, you must scan them and move the resulting files to the application repository. Electronic documents can come from various sources in various formats.

This tutorial examines different ways to place documents into your application for processing. You can set up a scanner for use with Datacap, which supports both ISIS and TWAIN scanners. For purposes of demonstration, it is assumed that you can set up a scanner with an ISIS driver attached to the computer that you are using. However, you can skip this requirement if you do not have a scanner with an ISIS driver. In that case, you can follow the virtual scanning examples.

For details about remote scanning with a TWAIN scanner, see [Remote scanning](#).

- [Electronic document input \(virtual scanning\)](#)
If your application is processing documents that are already available in electronic format, you can use *virtual scanning* to input the documents.
- [Hardcopy document scans](#)
Datacap supports local scanning and remote scanning.
- [TravelDocs: Batch creation with VScan](#)
Because sample electronic images are already installed, you do not need to scan paper documents. Instead, you can complete a virtual scan to create a batch.
- [Local scanner setup \(optional\)](#)
The default application framework does not include a scan task for scanning hardcopy documents into your application. Therefore, you must create a scan task for the TravelDocs application.

Parent topic: [Datacap application development](#)

Electronic document input (virtual scanning)

If your application is processing documents that are already available in electronic format, you can use *virtual scanning* to input the documents.

Datacap can manage a wide range of document types, including PDF files, fax files, and Microsoft Office documents. In addition, Datacap can ingest documents from various sources, including email and fax.

To scan image files from a shared folder on a network, or from a local folder, configure your application to use VScan actions. For details about VScan actions, see the Datacap Studio online help for the VScan action library.

The Datacap Studio application wizard generates an application framework that includes a virtual scanning task that copies files from the specified folder to the runtime batch folder. The virtual scan action is useful for application development and testing.

The Scan action copies the documents to the target location, and maintains the original files in the images folder.

- [Document conversion](#)
If your documents are not already in single-page TIFF format, you must convert them during the first stage of the processing workflow.

Parent topic: [Document input](#)

Related concepts:

[Exporting data](#)

[Datacap Connector actions](#)

Document conversion

If your documents are not already in single-page TIFF format, you must convert them during the first stage of the processing workflow.

The action categories in the Convert library can process various file types.

- Excel
- HTML
- Image files (JPEG, BMP, PNG, and GIF)
- Outlook
- PDF
- RTF
- Multi-page TIFF
- TXT
- Word
- ZIP - extracts image files that can then be converted by other conversion actions

Parent topic: [Electronic document input \(virtual scanning\)](#)

Hardcopy document scans

Datacap supports local scanning and remote scanning.

- Local scanning uses a scanner that is attached to and controlled from the Datacap Desktop component. Datacap Desktop supports ISIS and TWAIN scanners.
- Remote scanning uses the Datacap Web Client to scan and then upload the documents. The Datacap Web Client supports TWAIN scanners only.
- FastDoc scanning uses the FastDoc interface.

Confirm that your scanner driver is installed and that the scanner is functioning properly before you attempt to configure Datacap to use the scanner.

- [Local scanning](#)
When you scan from Datacap Desktop by using a local scanner, the scanned image files are delivered directly to the application's runtime batch folder. The scan task is responsible for creating the runtime batch files.
- [Remote scanning](#)
You can scan documents into a Datacap application by using the Datacap Web Client.

Parent topic: [Document input](#)

Related information:

[Starting Fastdocs](#)

Local scanning

When you scan from Datacap Desktop by using a local scanner, the scanned image files are delivered directly to the application's runtime batch folder. The scan task is responsible for creating the runtime batch files.

The application framework that is generated by the Datacap application wizard does not include a scan task. So, you need to create a scan task before you can scan locally. To create a scan task:

- Remove the existing VScan task from the Main Job workflow or create a new workflow for scanning (because a job can have only one batch creation task).
- Add a scan task to the workflow.
Tip: As an alternative to removing VScan and adding a scan task to the workflow, you can configure VScan to do physical scanning.
- Configure the scanner settings.
- Create a shortcut for the new scan task.

Detailed instructions are provided in [Local scanner setup \(optional\)](#). The instructions are specific to the TravelDocs application, but you can generalize them for any Datacap application.

Parent topic: [Hardcopy document scans](#)

Remote scanning

You can scan documents into a Datacap application by using the Datacap Web Client.

Remote scanning is typically a two-step process:

- Use a web scan task to scan the pages and save the image files locally.
- Use an upload task to upload the image files and runtime batch files to the application's batches folder.

The default application framework includes a web scan task. So, you do not need to create one. For more information, see [Datacap Web Client and remote scanning](#).

Parent topic: [Hardcopy document scans](#)

TravelDocs: Batch creation with VScan

Because sample electronic images are already installed, you do not need to scan paper documents. Instead, you can complete a virtual scan to create a batch.

- [Scanning the sample documents from the application images folder](#)
The VScan (virtual scanning) ruleset copies files from the images folder into the runtime batch folder of

the application.

- [Modifying the VScan ruleset](#)

Because the default VScan ruleset copies only the first four files, you must modify the VScan ruleset to copy up to 20 files.

- [Running VScan to generate a batch](#)

You can generate a batch in Datacap Studio by running the VScan task.

- [Examining the files in the runtime batch folder](#)

When you start a new batch, Datacap creates a runtime batch folder within the application batches folder.

Parent topic: [Document input](#)

Scanning the sample documents from the application images folder

The VScan (virtual scanning) ruleset copies files from the images folder into the runtime batch folder of the application.

About this task

The Datacap installation includes sample document images in the images folder of the application.

To work with the TravelDocs application, use the sample images that are provided, as summarized in the table.

Rental_Agreement	Optional_Insurance	Air_Ticket	Room_Receipt
Images_Page_01.tif	Images_Page_02.tif	Images_Page_06.tif	Images_Page_09.tif
Images_Page_03.tif	Images_Page_05.tif	Images_Page_07.tif	Images_Page_10.tif
Images_Page_04.tif		Images_Page_08.tif	Images_Page_11.tif

Parent topic: [TravelDocs: Batch creation with VScan](#)

Modifying the VScan ruleset

Because the default VScan ruleset copies only the first four files, you must modify the VScan ruleset to copy up to 20 files.

Procedure

1. On the Datacap Studio Rulemanager tab, Rulesets pane, select the VScan ruleset and click Lock/Unlock ruleset to lock the ruleset for editing.
2. Expand the VScan ruleset.
3. Select the SetMaxImageFiles action.
4. In the Properties pane, under Parameters, change the StrParam value from 4 to 20.
5. In the Rulesets pane, click Save, and then click Lock/Unlock ruleset and select Publish ruleset.

Parent topic: [TravelDocs: Batch creation with VScan](#)

Running VScan to generate a batch

You can generate a batch in Datacap Studio by running the VScan task.

Procedure

1. Click the Datacap Studio Test tab.
2. In the Workflow pane, select the VScan task profile under Main Job.
3. Click New to start a new batch.
4. Click Process rules for target object on the main Test tab toolbar.
5. When you are prompted to release the batch, click Advance. The Advance command moves the batch to the next step in the workflow, which in this case is PageID.
6. If the runtime batch hierarchy is not already visible, click the Runtime batch hierarchy tab. 11 pages are denoted as type `Other`, which is the default type that is assigned to all pages before page identification. Attention: If no pages are visible, ensure that you copied the sample image files as described in [Scanning the sample documents from the application images folder](#).
7. Right-click the running batch icon in the Workflow pane and select Cancel. This step cancels the running of the PageID task profile because you did not define the rules for page identification.

Parent topic: [TravelDocs: Batch creation with VScan](#)

Examining the files in the runtime batch folder

When you start a new batch, Datacap creates a runtime batch folder within the application batches folder.

About this task

The name of the folder matches the numeric batch identifier that Datacap generates automatically. In this example, `20100332.001` is the runtime batch folder.

```
C:\Datacap\TravelDocs\batches\20100332.001
```

Datacap stores all of the files that are associated with this batch in the runtime batch folder.

Procedure

1. Open the application's most recent batch folder (`C:\Datacap\TravelDocs\batches\<batch_id>`). The folder contains the following files:

File	Description
TM0000*.tif	A copy of each of the original sample image files (copied from the images folder).
VScan.scr ipt	A file to aid in debugging.
VScan.xml	The runtime document hierarchy that is generated by the VScan task profile.
Vscan_rrs.log	The log file that is generated by the VScan task profile. The log file contains detailed descriptions of all the actions that are started by the task profile and is useful for troubleshooting. For more information, see Datacap log files .
PageID.xml	A copy of the runtime document hierarchy ready for use by the next task profile in the workflow (PageID).

2. Open the VScan.xml file in any XML editor or text editor.

```
<?xml-stylesheet type="text/xsl"
href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_...\
\dco.xsl"?>
<B
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev130
_20100332.001">
  <V n="TYPE">TravelDocs</V>
  <V n="LAST_RR_TPROFILE">VScan:m:eRun</V>
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev130
_TM000001">
  <V n="TYPE">Other</V>
  <V n="STATUS">49</V>
  <V n="IMAGEFILE">tm000001.tif</V>
  <V n="ScanSrcPath">c:\datacap\traveldocs\images\images_page_01.tif</V>
</P>
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev130
_TM000002">
  <V n="TYPE">Other</V>
  <V n="STATUS">49</V>
  <V n="IMAGEFILE">tm000002.tif</V>
  <V n="ScanSrcPath">c:\datacap\traveldocs\images\images_page_02.tif</V>
</P>
etc.
```

The VScan.xml file, 20100332.001 contains the runtime batch ID. The file also indicates that a Page Type of *Other* is initially assigned to all pages. A *STATUS* of 49 indicates that the page scanned successfully.

3. Close the file.

Parent topic: [TravelDocs: Batch creation with VScan](#)

Local scanner setup (optional)

The default application framework does not include a scan task for scanning hardcopy documents into your application. Therefore, you must create a scan task for the TravelDocs application.

If you do not have a scanner that is attached to your computer, you do not need to set up a local scanner. If you have a scanner, ensure that the scanner driver is installed and that the scanner is working before you proceed.

You cannot include both a scan task and a virtual scan task in the same job workflow. You can have only one batch creation task per workflow. Because this tutorial requires a VScan task, you must copy the Main Job workflow, delete the VScan task from the copy, and create the scan task.

- [Creating the scan task in the Datacap Web Client](#)
You can use Datacap Web Client to create and configure the scan task.
- [Creating a shortcut for the new scan task](#)
To run the scan task by using the Scancl.aspx page, you must create a shortcut in Datacap Web Client. If you are using Datacap Desktop to run the scan task, a shortcut is not required.
- [Running the scan task](#)
Depending on how the task is configured, you can run the scan task in either Datacap Web Client or Datacap Desktop.

Parent topic: [Document input](#)

Creating the scan task in the Datacap Web Client

You can use Datacap Web Client to create and configure the scan task.

Procedure

To create the scan task follow this procedure.

1. Open the Datacap Web Client and log in to your application.
2. Click the Administrator tab and then click Workflow.
3. Select the Main Job workflow and click Copy.
4. Name the new job Scan Job, enter the Description such as `ISIS scan` or `TWAIN scan`, and click Apply.
5. Expand the new Scan Job, select the VScan task, click Remove and click OK.
Important: A job can have only one batch creation task. The VScan task and Scan tasks are both batch creation tasks so you must remove VScan.
6. Select Scan Job and click New to create a new task.
7. In the Selected task details section, enter or select these values for the new task.
 - o Name: `MyISISscan` or `MyTWAINscan`
 - o Description: `MyISISscan` or `MyTWAINscan`
 - o Mode: Batch Creation
 - o Queue to: None
 - o Store: None
 - o Program: Datacap Desktop
Attention: To complete remote scanning through the Datacap Web Client, select `Scancl.aspx`. You can select Multiple to configure both Datacap Desktop and `Scancl.aspx`.
8. Click Create Setup, and then click Setup.
9. Add another Datacap Desktop panel set of key and value fields. Enter the application name, such as `TravelDocs`, in the key field and in the value field, enter either `DotScanPanels.ISISScan` for ISIS scanners or `DotScanPanels.TWAINscan` for TWAIN scanners. Click Save.
10. Select the new task, such as `MyISISscan` or `MyTWAINscan`, and press Ctrl+Up Arrow to move the task to the top of the workflow.
11. Click Apply, and then click OK.

Parent topic: [Local scanner setup \(optional\)](#)

Creating a shortcut for the new scan task

To run the scan task by using the `Scancl.aspx` page, you must create a shortcut in Datacap Web Client. If you are using Datacap Desktop to run the scan task, a shortcut is not required.

Procedure

To create a shortcut for the new scan task:

1. In the Datacap Web Client, click the Administrator tab.
2. Click the Shortcuts tab, and click New to create a new shortcut.
3. In the Selected shortcut details section, enter or select these values for the following fields:
 - a. Name: `Scan`
 - b. Description: `Scan task`
 - c. Mode: Manual for Hold .
 - d. Under Permissions, clear the check boxes and click the `MyISISscan` check box under Scan Job.
4. Click Save.

Running the scan task

Depending on how the task is configured, you can run the scan task in either Datacap Web Client or Datacap Desktop.

Procedure

1. Load a page into your scanner's feeder.
2. If you are using remote scanning (scancl.aspx), complete the Scan task as follows:
 - a. On the Operations tab of the Datacap Web Client, click Scan.
 - b. After scancl.aspx loads into the Datacap Web Client, click Scan.
 - c. After Datacap scans the page, click OK and Done.
Important: To upload a scanned image to the Datacap server, you also must complete the Upload task, which you can start on the Operations tab.
3. If you are using Datacap Desktop, complete the Scan task as follows:
 - a. Start the Datacap Desktop program. On Windows in the Start menu click IBM Datacap Clients > Datacap Desktop.
 - b. Enter `TravelDocs` for the Application. Enter `admin` for the User and Password, and enter `1` for the Station. Click Login.
 - c. Select Scan from the Shortcut menu, and click Start.
 - d. For the first time that you are using Datacap Desktop with a scanner, or if you want to change the scanner that you used previously, click Select....
 - e. Choose the scanner that you want to use, and configure the scanner by completing one of the following tasks:
 - Set the values for functions, such as scan resolution, paper source, color mode, and others.
 - Click Configure and set the options in the scanner driver user interface.
 - f. In the Datacap Desktop main window, click Scan.

Page Identification

Page identification is one of the first steps in any Datacap application. All incoming pages are initially assigned the default page type Other. Before Datacap can assemble those pages into documents and extract data from the pages, it must determine the correct type for each page.

Page identification methods include fingerprint recognition, structure-based identification, text matching, and manual page identification. Image enhancement is typically done before page identification to remove lines, shading, and other graphic elements that might interfere with the recognition process.

- [Page identification methods](#)
Datacap supports several methods for page identification, which is also known as classification.
- [Image Enhancement](#)
Image enhancement consists of cleaning up images and removing elements that might produce recognition errors. You must complete any required image enhancement before page identification.
- [TravelDocs: Fingerprint library creation](#)
To create the initial fingerprint library for TravelDocs, you must change the default fingerprint creation method and create fingerprints for known page types.
- [TravelDocs: Sample fingerprint image enhancement](#)
To enhance the sample fingerprint images, you must determine the image-processing settings and apply them to the sample fingerprint files.

- [TravelDocs: Run a batch through the workflow](#)
After you create the initial fingerprint library and determine the appropriate image-processing settings, you can run a batch through the workflow.

Parent topic: [Datacap application development](#)

Page identification methods

Datacap supports several methods for page identification, which is also known as classification.

Page identification includes the following methods.

- Fingerprint matching
- Structure-based identification
- Text matching
- Category-based classification
- Rule-based documentation
- Manual page identification

Additionally, if your application supports only a single-page type, you can assign a static page type to all incoming pages.

- [Fingerprint matching](#)
Fingerprint matching is the method of identifying a page type by using fingerprints. Specifically, a page's fingerprint is compared to the fingerprints in a database in which each database fingerprint has an associated page type. As a result of fingerprint matching, a page is assigned the page type of the most closely matching database fingerprint.
- [Structure-based page identification](#)
Structure-based identification uses the position of a page within the batch to determine its type.
- [Text matching](#)
To complete page identification by using text matching, you must first complete a full page recognition. You can then search the recognition results for a string that is unique to each page type.
- [IBM Content Classification: Category and rule-based classification](#)
The Datacap CC actions use IBM® Content Classification technology for page type identification in the following interrelated ways: category-based classification and rule-based classification.
- [Manual page identification](#)
Although many page identification techniques identify pages automatically, you can configure your application to display unrecognized pages to an operator for manual identification.

Parent topic: [Page Identification](#)

Fingerprint matching

Fingerprint matching is the method of identifying a page type by using fingerprints. Specifically, a page's fingerprint is compared to the fingerprints in a database in which each database fingerprint has an associated page type. As a result of fingerprint matching, a page is assigned the page type of the most closely matching database fingerprint.

A fingerprint is a representation of either the relative densities of different regions of the page (an image-based fingerprint) or the location of text on the page (an OCR-based fingerprint). For more information, see the *Selecting the fingerprint creation mode* section in this topic.

For example, assume that the fingerprint for an incoming page matches the Hotel #1 room receipt fingerprint. Datacap assigns the page type called Room_Receipt. It then records the ID of the matching fingerprint in the

runtime batch hierarchy. The match is not exact because the data on the page is most likely different. However, you are just looking for the best match possible.

Selecting the fingerprint creation mode

Datacap provides two primary methods for generating page fingerprints.

Image analysis

This method scans the page image to identify the composite blackness of different regions of the page. This method provides fast page identification, but it requires that you do recognition later.

Full page recognition

This method does optical character recognition to identify the locations of text within the page. This method takes longer, especially with pages that include handwritten text. However, it reduces the time from subsequent workflow tasks because the full page recognition results are available for use.

Both of these methods write the resulting information to a CCO file that is stored with the original TIFF image file in the fingerprint folder for the application.

Remember: The method that you use for creating library fingerprints must be the same as the method that you use to generate runtime fingerprints during page identification.

For example, if you decide to use image analysis, you must use image analysis in both the FingerprintAdd and PageID rulesets.

Important: Do not try to combine these methods because the recognition results are probably not accurate.

Image analysis

Image analysis uses a pixel-based algorithm to generate a CCO fingerprint file that represents the relative blackness of different regions of the page.

The AnalyzeImage action in the Recog_Shared actions library does image analysis on an image file.

Library	Action	Description
Recog_Shared	AnalyzeImage	Converts the TIFF image file that represents the current page to a CCO fingerprint file.

Full page recognition

Full page recognition, as the name suggests, uses the text and location of text on the page to generate the CCO fingerprint file. Datacap includes three optical character recognition (OCR) engines, plus one intelligent character recognition (ICR) engine that you can use to do full page recognition:

OCR_A

FineReader OCR engine.

OCR_S

Nuance (formerly ScanSoft) OmniPage OCR engine.

OCR_SR

Newer implementation of the Nuance OmniPage OCR engine.

ICR_C

Open Text RecoStar ICR engine.

Other ICR engines are available as options. As a rule, the OCR engines work well with machine-printed text, whereas the ICR engine works well with hand-printed and machine-printed text.

Datacap includes actions libraries for each recognition engine (OCR_A, OCR_S, OCR_SR, and ICR_C). Each library includes its own version of the full page recognition action.

Library	Action	Description
ocr_a	RecognizePageOCR_A	Recognizes all characters on the current page and populates CCO fingerprint file for the page with the recognition results.
OCR_s	RecognizePageOCR_S	Recognizes all characters on the current page and populates the CCO fingerprint file for the page with the recognition results.
ocr_sr	RecognizePageOCR_S	Recognizes all characters on the current page and populates the CCO fingerprint file for the page with the recognition results.
icr_c	RecognizePageICR_C	Recognizes all characters on the current page and populates the CCO fingerprint file for the page with the recognition results.

Fingerprint matching actions

Here are some actions that are involved in fingerprint matching:

Library	Action	Description
AutoDoc	FindFingerprint	Tries to match the current page fingerprint to a fingerprint in the application fingerprint library.

Parent topic: [Page identification methods](#)

Structure-based page identification

Structure-based identification uses the position of a page within the batch to determine its type.

You can assign page types that are based on position when application manages only one page type, or when the document structure is consistent. For example, all documents are two pages with a main page and a trailing page. For structure-based identification, use the Set Page Type action.

Library	Action	Description
DCO	SetPageType	Assigns a page type to the current page.
DCO	SetPageStatus	Sets the status of the current page.

If a batch contains documents of varying length, you can use separator pages between documents. For an example that uses barcoded separators, look at the Datacap Accounts Payable (APT) foundation application that you can run with Datacap.

When you identify a page by using structure-based identification, the page is not matched to a fingerprint. Therefore, even though recognition zones are available for your application to locate data during recognition, the zones are not aligned to the scanned image. After you identify a page with structure-based methods, the application can be customized to call CreateFields. When this call is in place, recognition zones are located wherever they were defined on the original fingerprint image for that page type. The zone locations are not adjusted for shifting of the scanned image as they would be if Fingerprint matching were used. However, this

limitation can be mostly overcome in at least two ways. You can crop and de-skew the image during an image-processing step. You can use pattern-match anchors to align the zones.

Parent topic: [Page identification methods](#)

Text matching

To complete page identification by using text matching, you must first complete a full page recognition. You can then search the recognition results for a string that is unique to each page type.

In the TravelDocs application, the first function attempts a full page recognition and searches for the string `Pickup` on the current page. If the function finds `Pickup`, it assigns the page type `Rental_Agreement`. If the function does not find `Pickup`, it fails, and the second function searches for the string `Flight`. If the second function finds `Flight`, it assigns the page type `Air_Ticket`. If it does not find `Flight`, the second function fails, and the third function searches for the string `Room`. If the third function finds `Room`, it assigns the page type `Room_Receipt`. If it does not find `Room`, the page remains with the page type `Other`.

As with the structure-based techniques, when you identify a page by using text matching, the page is not matched to a fingerprint. Therefore, even though recognition zones are available for your application to locate data during recognition, the zones are not aligned to the scanned image. After you identify a page with text-matching methods, you can customize the application to call `CreateFields`. This call locates the recognition zones where they were defined on the original fingerprint image for that page type. The zone locations are not adjusted for shifting of the scanned image in the same manner that Fingerprint matching can adjust locations. However, you can work around this limitation by using either of two methods: You can crop and de-skew the image during an image-processing step, or you can use pattern-match anchors to align the zones.

Parent topic: [Page identification methods](#)

IBM Content Classification: Category and rule-based classification

The Datacap CC actions use IBM® Content Classification technology for page type identification in the following interrelated ways: category-based classification and rule-based classification.

For information about IBM Content Classification, see [Classification overview](#).

Category-based classification

Category-based classification is a method of identifying the type of page text (or other text) by using an IBM Content Classification knowledge base. The text in question is compared against the text categories in the knowledge base to find the most closely matching category.

The following terms are helpful for understanding category-based classification:

<i>category confidence score</i>	The degree of similarity between a piece of text and an IBM Content Classification category that describes text. This similarity is expressed in a range from 0.0 - 1.0 with 1 indicating a perfect match. For example, the confidence score for a piece of text and a matching category might be 0.7.
<i>minimum category confidence score</i>	The minimum category confidence score that is required for a category to be considered a match. This minimum score is configurable.
<i>text confidence level</i>	The confidence score between a page's text and the closest matching category.

Rule-based classification

Rule-based classification is a method of identifying a page type by using rules that are defined in an IBM Content Classification decision plan. For example, a decision plan that is called *Mortgage* might have the following rule: “If the document contains the word ‘Loan’, create a field that is called ‘MyType’ with the value ‘Mortgage’”.

In Datacap, you use actions to specify the decision plan to run and the fields from the decision plan to save in the DCO page object. For example, you might call the following actions:

```
SetDecisionPlanCC("Mortgage")           // Specify the decision plan to use
SetDecisionPlanFieldsCC("MyType")       // Specify the fields to be set in the DCO
page object
RunDecisionPlanCC()                     // Run the decision plan
rrSet("MyType", "Page type")           // Copy the MyType field value as the page
type
```

Parent topic: [Page identification methods](#)

Manual page identification

Although many page identification techniques identify pages automatically, you can configure your application to display unrecognized pages to an operator for manual identification.

For more information, see [Adding a function for manual page identification](#).

Parent topic: [Page identification methods](#)

Image Enhancement

Image enhancement consists of cleaning up images and removing elements that might produce recognition errors. You must complete any required image enhancement before page identification.

- [Goal of image enhancement](#)
The goal of image enhancement is to eliminate lines, shading, misalignment, and other artifacts that can interfere with the recognition process.
- [When to complete image enhancement](#)
You complete image enhancement on fingerprint images when you are setting up the fingerprint library. You must complete it again on your document images after input but before page identification.

Parent topic: [Page Identification](#)

Goal of image enhancement

The goal of image enhancement is to eliminate lines, shading, misalignment, and other artifacts that can interfere with the recognition process.

The Datacap Image Processing tool provides a set of image enhancement capabilities that you can configure to handle various problem types. However, finding the best combination of image enhancement settings can take time, especially if your application must handle multiple page types. Image enhancement is typically done before the page type is known, in other words, before page identification. You must set up the image-processing properties in a way that works well for all page types.

Important: Some special cases are exceptions to this statement. Rules, or ImageFix Actions, can do multiple passes of image enhancement, which is also known as image processing, before or after page identification.

The rules can use different settings for different page types or based on other criteria. The most common use case is a single pass before page identification. Any image enhancement that is completed before fingerprint matching must be identical to the image enhancement that was done on the fingerprint template image when the template was created.

The default image-processing properties are designed to work well with typical printed pages that use plain black text on a white background. Establishing settings that work well for the pages your application must handle requires experimentation. For more information, see [Determining appropriate image-processing settings](#).

The settings that you establish are stored in the file `imagefix.ini` in the application's `dco_<application_name>` folder.

Parent topic: [Image Enhancement](#)

When to complete image enhancement

You complete image enhancement on fingerprint images when you are setting up the fingerprint library. You must complete it again on your document images after input but before page identification.

When you add fingerprints to the fingerprint library, Datacap queries whether to enhance the image. Typically, you experiment to find settings that work well for all page types. You can skip image enhancement initially, and then return and enhance the fingerprint images, after you determined appropriate settings. See [Applying new image-processing settings to enhance the fingerprint images](#).

After document input, you use the ImageFix ruleset to apply the same image-processing settings for image enhancement. The default ImageFix ruleset includes two rules.

- The first rule (ImageFix Load Settings) reads the image-processing properties from the settings (.ini) file.
- The second rule (Enhance Image) completes image processing on each page by using those settings and creates a backup of the original with a .tio extension.

Parent topic: [Image Enhancement](#)

TravelDocs: Fingerprint library creation

To create the initial fingerprint library for TravelDocs, you must change the default fingerprint creation method and create fingerprints for known page types.

- [Changing the fingerprint creation method](#)
The application framework that is generated by the Application wizard uses the image analysis method for fingerprint creation. All of the pages in the TravelDocs application are printed from a computer. So, you must convert the application to use full page recognition with the OCR_SR engine.
- [Fingerprint creation for known page types](#)
To create fingerprints for known page types, you must create fingerprint classes and add individual fingerprints.

Parent topic: [Page Identification](#)

Changing the fingerprint creation method

The application framework that is generated by the Application wizard uses the image analysis method for fingerprint creation. All of the pages in the TravelDocs application are printed from a computer. So, you must

convert the application to use full page recognition with the OCR_SR engine.

About this task

To change the fingerprint creation method, you must edit two of the rulesets that are defined on the Datacap Studio Rulemanager tab. The FingerprintAdd ruleset runs whenever you add a fingerprint to the fingerprint library. PageID generates the runtime fingerprints and matches them to determine the type of each incoming page. You can modify these rulesets to complete full page recognition instead of image analysis.

Procedure

To modify the FingerprintAdd and Page ID rulesets:

1. On the Datacap Studio Rulemanager tab, in the Rulesets pane, select the FingerprintAdd ruleset and click Lock/Unlock ruleset (padlock) to lock the ruleset for editing.
2. Expand the FingerprintAdd ruleset completely.
3. Right-click the AnalyzeImage action and choose Remove.
4. Click the Actions library tab.
5. Expand the OCR_SR library and select RecognizePageOCR_S.
6. Make sure FingerprintAdd: Other Function 1 is selected in the Rulesets pane.
7. Click Add to function.
8. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and select Publish ruleset.
9. Select the PageID ruleset and click Lock/Unlock ruleset to lock the ruleset for editing. Then, expand the ruleset and the PageID rule.
10. Remove the AnalyzeImage action and replace it with the RecognizePageOCR_S action. If necessary, use Up arrow or Down arrow to move the action to the correct position within the function.
11. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and select Publish ruleset.

Parent topic: [TravelDocs: Fingerprint library creation](#)

Fingerprint creation for known page types

To create fingerprints for known page types, you must create fingerprint classes and add individual fingerprints.

- [Creating fingerprint classes](#)
By using classes, you can categorize fingerprints within your application.
- [Adding individual fingerprints](#)
After you create the fingerprint classes, you add them to the fingerprint library.

Parent topic: [TravelDocs: Fingerprint library creation](#)

Creating fingerprint classes

By using classes, you can categorize fingerprints within your application.

About this task

The default framework includes two classes:

<Global>

This class includes the generic 555 fingerprint with page type `Other`. The generic fingerprint is useful because it enables application development without page fingerprints.

<New>

When you use the FindFingerprint action during page identification, you can create fingerprints automatically for unrecognized pages. If you call FindFingerprint with the parameter True and Datacap does not find a matching fingerprint, Datacap adds the runtime fingerprint to the `New` class.

Here, you create a class for each document type: `Car_Rental`, `Hotel`, and `Flight`. Categorization by document type is not required but provides a useful way to organize many fingerprints.

Procedure

To create the fingerprint classes:

1. On the Datacap Studio Zones tab, in the Fingerprints pane, click Add new item and select Add fingerprint class.
2. Enter `Car_Rental` and click OK.
3. Repeat for `Flight` and `Hotel`.

Parent topic: [Fingerprint creation for known page types](#)

Adding individual fingerprints

After you create the fingerprint classes, you add them to the fingerprint library.

Procedure

To add individual fingerprints:

1. In the Fingerprints pane, right-click the new `Car_Rental` class and choose Add fingerprint.
2. Browse to the folder where the TravelDocs fingerprint images are located.
3. Select `Car1.tif`, and click Open. When you are prompted to enhance the image, click No (you enhance the image later). Datacap requires a few minutes to add the new fingerprint.
4. Repeat to add `Car2.tif`, `Car3.tif`, `Car4.tif`, `Car5.tif`, and `Car6.tif`. Again, do not enhance the images.
5. In the Fingerprints pane, select the first car rental fingerprint and confirm that it is a rental agreement page. Then, click Type at the top of the pane and choose `Rental_Agreement`.
6. Repeat to assign page types to the remaining car rental fingerprints. Use `Rental_Agreement` for the rental agreement pages and `Optional_Insurance` for the optional insurance pages.
7. Add `Flight1.tif`, `Flight2.tif`, and `Flight3.tif` to the `Flight` class and assign the type `Air_Ticket`.
8. Add `Hotel1.tif`, `Hotel2.tif`, and `Hotel3.tif` to the `Hotel` class and assign the type `Room_Receipt`.
Attention: Do not add `Hotel4.tif` or `Hotel5.tif`.

Parent topic: [Fingerprint creation for known page types](#)

TravelDocs: Sample fingerprint image enhancement

To enhance the sample fingerprint images, you must determine the image-processing settings and apply them to the sample fingerprint files.

- [Determining appropriate image-processing settings](#)
Because image enhancement is completed before page identification, you must set up the image-processing properties for all page types.
- [Applying new image-processing settings to enhance the fingerprint images](#)
After you determined appropriate image-processing settings, you can apply these settings to all of the

sample fingerprint files.

Parent topic: [Page Identification](#)

Determining appropriate image-processing settings

Because image enhancement is completed before page identification, you must set up the image-processing properties for all page types.

About this task

Most Datacap applications must manage multiple page types.

Important: Some special cases are exceptions to this statement. Rules complete multiple passes of image enhancement before or after page identification by using different settings for different page types or based on other criteria. The most common use case is a single pass before page identification. Any image enhancement that is completed before fingerprint matching must be identical to the image enhancement completed on the fingerprint template image when the template was created.

The default image-processing properties are designed to work with typical printed pages that use plain black text on a white background. One of the sample air ticket pages contains white text on a black background, which is the most difficult page to process. This procedure addresses first the page with white text on a black background.

Procedure

1. In the Fingerprints pane on the Zones tab, expand the Flight class and select the third fingerprint (Airline #3).
2. In the Image View pane, click Open image processing settings in the upper right.
3. Click Run image processing to apply the default image-processing properties, as defined in the Properties pane.
4. Click Reset image to revert to the original image.
5. In the Properties pane, change the settings as follows:

Category	Property	Default setting	New setting
Border Removal	Border Removal	True	False
Inverse Text Correction	Minimum Area Width	300	100
Line Removal	Minimum Length	50	30

6. Click Save and choose Save settings. Then, click OK.
Attention: When you save the settings, Datacap saves the new image enhancement properties in the file C:\Datacap\TravelDocs\dco_TravelDocs\imagefix.ini. Datacap uses the same settings file for the image processing that takes place before page identification (ImageFix).
7. Click Run image processing to apply the new image-processing properties. This time all of the vertical and horizontal lines disappear. The top of the page is not clipped, and the white text on a black background is converted to black text on a white background.
8. Close the Image Processing window without saving the enhanced image.
9. Next, in the Fingerprints pane, select the second air ticket fingerprint (Airline #2). There are problems on this page with the default settings, but you can try it with the new settings.
10. In the Image View pane, click Open image processing settings in the upper right.
11. Click Run image processing to apply the new image-processing properties.

The horizontal lines are removed while everything else is intact.

12. Close the Image Processing window without saving the enhanced image.

Parent topic: [TravelDocs: Sample fingerprint image enhancement](#)

Applying new image-processing settings to enhance the fingerprint images

After you determined appropriate image-processing settings, you can apply these settings to all of the sample fingerprint files.

Procedure

To apply the appropriate processing settings to the sample fingerprint files:

1. In the Fingerprints pane, expand the Car_Rental class and select the first Rental_Agreement fingerprint.
2. In the Image View pane, click Open image processing settings.
3. Click Run image processing to apply the image-processing properties.
4. Click Save, choose Save image, and click OK. Then, click x to close the Image Processing window.
5. Repeat to apply the same image-processing properties to all of the other fingerprints. Make sure that you explicitly save each image after image processing.

Parent topic: [TravelDocs: Sample fingerprint image enhancement](#)

TravelDocs: Run a batch through the workflow

After you create the initial fingerprint library and determine the appropriate image-processing settings, you can run a batch through the workflow.

In summary, you completed these tasks in developing your TravelDocs application.

- Created an application framework for the TravelDocs application by using the Datacap Studio Application wizard.
- Modified the default document hierarchy to include the document types and pages types the TravelDocs application supports.
- Specified the required structure for documents and pages within a batch according to the business requirements.
- Within the document hierarchy, defined the fields of interest for each page type.
- Created the initial fingerprint library by using one sample image for each known variant of each page type.

In terms of implementing the workflow. You did not attach any rules to the document hierarchy, though some default rules are attached to the default elements. However, you can run a batch through the PageID task to make sure that the application is handling page identification correctly.

- [Processing a batch](#)
For testing purposes, you can process a batch on the Test tab of Datacap Studio.
- [Runtime batch folder contents](#)
The application's most recent batch folder is at C:\Datacap\TravelDocs\batches*< batch_identifier >*.
- [Checking the confidence levels on the runtime pages](#)
During page identification, Datacap assigns a confidence level to each page. This process indicates the degree of similarity between the runtime page and the fingerprint that matches the page most closely.

Parent topic: [Page Identification](#)

Processing a batch

For testing purposes, you can process a batch on the Test tab of Datacap Studio.

Procedure

To process a batch:

1. Open Datacap Studio and click the Test tab.
2. In the Workflow pane, select the VScan task profile under Main Job.
3. Click New to start a new batch.
4. Click Process rules for target object on the main Test tab toolbar.
5. When asked if you want to release the batch, click Advance. The batch is moved to the next step in the workflow (PageID).
6. Click Process rules for target object on the main Test tab toolbar and wait while the task profile runs. It might take a few moments because Datacap must do full page OCR on all the images in the batch.
7. When asked if you want to release the batch, click Advance. The batch is moved to the next step in the workflow (Profiler).
8. On the Runtime batch hierarchy tab, scroll through the list to see the page types that are assigned to TM000001, TM000002, and so on.
9. Right-click the running batch button in the Workflow pane and choose Cancel. You do not run the Profiler task profile until you assign rules.

Parent topic: [TravelDocs: Run a batch through the workflow](#)

Runtime batch folder contents

The application's most recent batch folder is at C:\Datacap\TravelDocs\batches*< batch_identifier >*.

The runtime batch folder contains these files.

File	Description
TM00000*.tif	An image-enhanced version of each of the sample image files.
TM00000*.tio	A copy of each of the original image files.
TM00000*c.xml	The results of full page recognition for each image file.
TM00000*.cco	The fingerprint file for each of the image files.
PageID.xml	The runtime document hierarchy that is generated by the PageID task profile.
pageid_rrs.log	The log file that is generated by the PageID task profile.
VScan.xml	The runtime document hierarchy that is generated by the VScan task profile.
vscan_rrs.log	The log file that is generated by the VScan task profile.
Profiler.xml	A copy of the runtime document hierarchy ready for use by the next task profile in the workflow (Profiler).

Parent topic: [TravelDocs: Run a batch through the workflow](#)

Checking the confidence levels on the runtime pages

During page identification, Datacap assigns a confidence level to each page. This process indicates the degree of similarity between the runtime page and the fingerprint that matches the page most closely.

About this task

You can see the confidence level for each page in the runtime batch file (PageID.xml) that is generated by the PageID task profile.

Procedure

To check the confidence levels on the runtime pages:

1. Open the application's most recent batch folder (C:\Datacap\TravelDocs\batches*<batch_identifier>*).
2. Open the file PageID.xml in an XML viewer or in Notepad. This file includes the confidence level that is assigned to each page in the batch, and the identifier of the matching fingerprint.

```
<?xml-stylesheet type="text/xsl"
href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_...\
\dco.xsl"?>
<B
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev035
_20100321.002">
  <V n="TYPE">TravelDocs</V>
  <V n="LAST_RR_TPROFILE">PageID:m:eRun</V>
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev035
_TM000001">
  <V n="TYPE">Rental_Agreement</V>
  <V n="STATUS">49</V>
  <V n="IMAGEFILE">tm000001.tif</V>
  <V
n="ScanSrcPath">c:\datacap\traveldocs\images\images_page_01.tif</V>
  <V n="RecogStatus">0</V>
  <V n="Confidence">0.9727517</V>      <-- Confidence level
  <V n="Image_Offset">0,0</V>
  <V n="TemplateID">556</V>          <-- identifier of matching
fingerprint
  <V n="Fingerprint Created">No</V>
  </P>
  ...
```

Results

The default application framework uses a confidence threshold of 0.7, so anything above 0.7 is considered a match. In the example above, the confidence level is 0.97, so the page is a good match. If multiple fingerprints match with a confidence level above 0.7, Datacap selects the fingerprint with the highest confidence value. Important: The SetProblemValue action in the PageID: Set Fingerprint Params rule specifies the minimum required confidence level.

Parent topic: [TravelDocs: Run a batch through the workflow](#)

Rule Execution

Rule execution refers to how rules are associated with specific objects in the document hierarchy and how Datacap processes a batch of documents.

The Datacap workflow moves batches through the workflow from task to task. Task profiles are implemented as rulesets, which are constructed by using rules and actions.

You can use the Datacap Studio debugging tools to step through the PageID task profile and see how Datacap runs the rules.

A ruleset consists of one or more rules that contain functions and actions. When you run a rule, you are running the functions and actions that are in them.

The following example describes the execution flow that allows actions to run by using an `if, then, else` programming model.

```
Run a rule:
for each function in the rule
  for each action in the function
    run the action
    if the action returns false, exit from this function
  next action in the function
  if the last action in the function returns true, exit from this rule
next function in the rule
if the last action that is run returns true, then the rule result is true,
else the rule result is false
For validation rules, true means that validation passed, false means that it failed
```

The actions that are frequently used to control rules execution flow are `rrCompare`, `rrCompareNot`, `GoToNextFunction`, and `SetReturnValue`.

- [Association of rules with objects](#)
Rules run when they are assigned to specific objects in the document hierarchy, and only if the parent rulesets are included in the current task profile.
- [Order of rule execution](#)
Rules are run according to the position of the ruleset within the task profile, and the position of the associated object within the document hierarchy.
- [Running rules directly on images by using Datacap Web Services transactional endpoints](#)
The transactional endpoints can be used when you want to invoke a Datacap Web Services service by making REST calls. You provide input files such as images and execute a set of functions with rules that does image processing , OCR, redaction, and so on.
- [TravelDocs: Stepping a batch through the PageID task profile](#)
Even though you did not add any new functions to the application, you can run the VScan and PageID rulesets again and step through the actions.

Parent topic: [Datacap application development](#)

Association of rules with objects

Rules run when they are assigned to specific objects in the document hierarchy, and only if the parent rulesets are included in the current task profile.

Datacap can run rules on batches, documents, pages, and fields.

- [Example 1: Batch-level rule execution](#)
When a rule assembles individual pages into different document types, the rule must run at the batch level.
- [Example 2: Page-level rule execution](#)
When a rule locates field zones on different pages, the rule must be run at the page level.

Parent topic: [Rule Execution](#)

Example 1: Batch-level rule execution

When a rule assembles individual pages into different document types, the rule must run at the batch level.

In the default Profiler task profile, the CreateDocs ruleset includes a rule that is called Create Docs. This rule assembles individual pages into documents that are based on the structure in the document hierarchy. For example, the TravelDocs car rental document type has a rental agreement page and an insurance coverage page.

From the object general information, you can observe these details about the pages.

Tip: To see the object general information with the rules for each page type within a document, lock the Document Hierarchy, right-click the page type, and select Manage variables.

1. The rental agreement page is required (`Min=1`), and that it is always the first page (`Order=1`).
2. There can be only one rental agreement page within each car rental document (`Max=1`).
3. The insurance page is optional (`Min=0`).

The Create Docs rule must run at the batch level because it assembles multiple document types.

When the Create Docs rule encounters a page of type `Rental_Agreement`, it creates a new document in the runtime batch hierarchy. If a rental agreement page is followed immediately in the batch by an `Optional_Insurance` page, Datacap adds the insurance page to the same car rental document. Otherwise, Datacap creates a new document.

Parent topic: [Association of rules with objects](#)

Example 2: Page-level rule execution

When a rule locates field zones on different pages, the rule must be run at the page level.

The default Recognize ruleset includes a rule that is called Recognize Page. This rule locates each field zone on the current page by using the positional information in the document hierarchy. The rule then uses OCR to get the data within each zone.

The rule must run at the page level because of these reasons:

- The fields are different for each page type (for example, the TravelDocs rental agreement page and the optional insurance page have different fields).
- The field positions are different for each variation of the page type. For example, the `Car_Type` field is at a different location on the page for each rental car company.

Tip: To see the variable information for any field, lock the Document Hierarchy, right-click the field, and choose Manage variables.

The position variables define the position of the field for each of the car rental agreement forms that are identified during fingerprinting. 678, 695, and 696 are the fingerprint IDs.

The Document Hierarchy shows that the Recognize Page rule runs at the page level for each page type.

The rule is not assigned to the page type `Other` because no fields are defined for `Other`.

Parent topic: [Association of rules with objects](#)

Order of rule execution

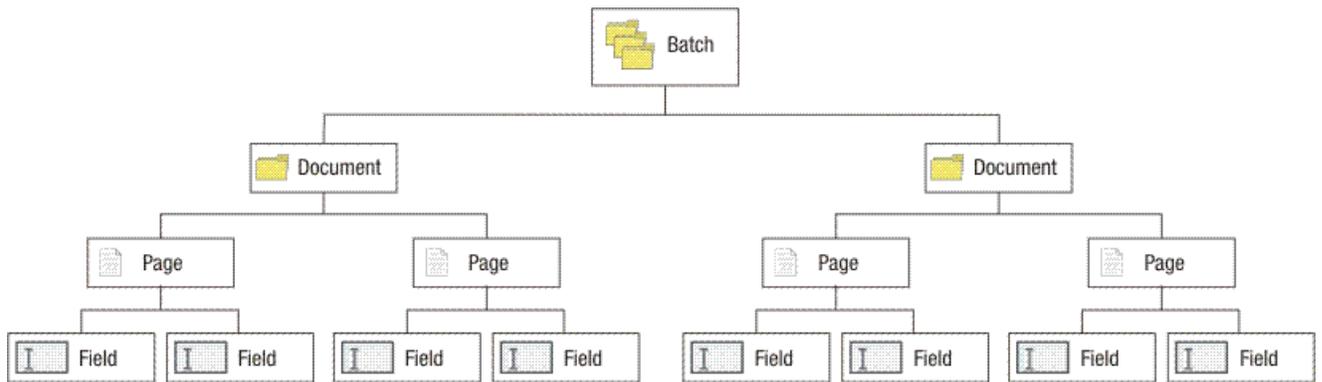
Rules are run according to the position of the ruleset within the task profile, and the position of the associated object within the document hierarchy.

Two conditions determine whether a rule is run. A rule runs only when it is associated with a specific object in the document hierarchy. Also, the rule runs only when the parent ruleset is included in the current task profile.

The position of a rule within its ruleset does not affect when the rule is run.

When you run a task profile that has more than one ruleset, Datacap runs the rulesets in the order that the Task Profiles pane displays them in Datacap Studio. For example, the PageID task profile, Datacap first runs the ImageFix ruleset, and next runs the PageID ruleset.

When Datacap runs a ruleset on a batch, it goes through the runtime batch hierarchy iteratively. (Page 1 of Document 1, including the two fields, is processed in its entirety before Datacap proceeds to Page 2 of Document 1.)



For each object, Datacap runs any rule in the current ruleset that is included in the object's *Open* element. For example, if you are running the Recognize ruleset on a page of type `Rental_Agreement`, Datacap runs the Recognize Page rule.

- Car_Rental
- Open
 - Rental_Agreement
 - Open
 - (global)
 - CreateDocs:Create fields
 - Recognize:Recognize Page

Restriction: You can include only one rule from a ruleset in an object's *Open* element.

The following example shows a portion of the document hierarchy for the TravelDocs application. Each object also has a *Close* element. Datacap runs rules in the object's *Close* element when it exits from the object. For objects at the lowest level of the hierarchy, *Close* rules run immediately after *Open* rules. For other objects, the *Close* rules do not run until Datacap processes the lower-level objects.

TravelDocs (Batch)

- Open
 - (global)
 - VScan: VScan
 - ImageFix: ImageFix Load Settings
 - PageID: Set Fingerprint Params
 - CreateDocs: Create Docs
 - Document Integrity: Batch Document
 - Export: Set Export Params

- Other (Page)
 - Open
 - (global)
 - ImageFix: Enhance Image
 - PageID: PageID
 - FingerprintAdd: FingerprintAdd
 - Close
- Car_Rental (Document)
 - Open
 - Rental Agreement (Page)
 - Open
 - (global)
 - CreateDocs: Create Fields
 - Recognize: Recognize Fields
 - Validate: Validate Page
 - Routing: Routing Rule 1
 - Export: Export Page Fields
 - Vendor (Field)
 - Pickup_Date (Field)
 - Open
 - Global
 - Validate: Validate Date
 - Close

The `TravelDocs Batch` rules run when batch processing begins, depending on the ruleset that you are running. For example, if you are running the `VScan` ruleset, Datacap runs the `VScan` rule.

`Other page` rules run each time Datacap begins processing a page of type `Other`, which is the default page type that is initially assigned to all pages. For example, if you are running the `PageID` ruleset, Datacap runs the `PageID` rule, which assigns the correct type to each page.

For the `Car_Rental` document, there are no document-level rules in this example.

The `Rental_Agreement page` rules run each time Datacap begins processing a page of type `Rental_Agreement`. For example, if you are running the `CreateDocs` ruleset, Datacap starts the `Create Fields` rule, which creates a page data file. Datacap uses the page data file to store the data that is captured later by the `Recognize` ruleset.

The `Pickup_Date field` rule runs when Datacap is processing a page of type `Rental_Agreement` and encounters a field of type `Pickup_Date`.

- [Example 1: Page identification rules](#)
A page identification rule identifies each incoming page by comparing the page image to the known page types and by using fingerprint recognition.
- [Example 2: Validation rules](#)
Validation rules are used to confirm that a field value conforms to a supported format.
- [Summary of order of rule execution](#)
The order of rule execution closely follows the document hierarchy, starting with the batch, proceeding to the document, and continuing down to the field level.

Parent topic: [Rule Execution](#)

Example 1: Page identification rules

A page identification rule identifies each incoming page by comparing the page image to the known page types and by using fingerprint recognition.

The PageID task profile includes a ruleset that is called PageID, which includes a rule that is called PageID.

The Sync DCO View with Ruleset View button shows which documents, pages, or fields are associated with the selected rule. For example, if you select the PageID rule and click Sync DCO View with Ruleset View, Datacap Studio expands the document hierarchy to show you the objects that are associated with the PageID rule.

In TravelDocs, the PageID rule is associated only with the page type `Other`. In this case, the rule runs whenever Datacap processes a page of type `Other` while it runs the PageID task profile. Datacap assigns the `Other` page type to all incoming pages as a default because all pages are initially unknown. Datacap runs this rule on all pages during page identification and, assuming the page matches one of the known types, assigns the correct type (for example, `Rental_Agreement` or `Air_Ticket`).

Parent topic: [Order of rule execution](#)

Example 2: Validation rules

Validation rules are used to confirm that a field value conforms to a supported format.

The Profiler and Verify task profiles both include the Validate ruleset. In the TravelDocs application, the Validate ruleset includes a rule that is called Validate Date, which returns `True` if a field value conforms to one of the supported date formats.

If you select the Validate Date rule and click Sync DCO View with Ruleset View, Datacap Studio expands the document hierarchy to show you which objects are associated with the rule.

In the TravelDocs application, the Validate Date rule is associated with the `Arrival_Date` field and `Departure_Date` field on pages of type `Room_Receipt`. This rule runs whenever Datacap processes an `Arrival_Date` field or `Departure_Date` field on a page of type `Room_Receipt` while it runs the Profiler or Verify task profile.

Parent topic: [Order of rule execution](#)

Summary of order of rule execution

The order of rule execution closely follows the document hierarchy, starting with the batch, proceeding to the document, and continuing down to the field level.

The following pseudocode describes the order of rule execution.

```
for each ruleset in the current task profile
  run batch-level "Open" rules
  for each document type
    run document-level "Open" rules
    for each page type in the current document type
      run page-level "Open" rules
      for each field in the current page type
        run field-level "Open" rules
        run field-level "Close" rules
      next field
    run page-level "Close" rules
  next page
  run document-level "Close" rules
next document
```

```
run batch-level "Close" rules
next ruleset
```

Parent topic: [Order of rule execution](#)

Running rules directly on images by using Datacap Web Services transactional endpoints

The transactional endpoints can be used when you want to invoke a Datacap Web Services service by making REST calls. You provide input files such as images and execute a set of functions with rules that does image processing, OCR, redaction, and so on.

About this task

You can run the images through whatever function Datacap rules can provide without needing to go through a workflow or series of tasks, and without using Datacap Server to queue and run the tasks.

Two potential use cases for transactional endpoints include:

- Extracting data from documents by uploading a new page file for the transaction, performing recognition and data extraction, getting the data file back, and parsing it for field values.
- Validating data by uploading a new page file for the transaction, uploading one or more data files with field values, running the validation rules, getting the updated page and data files back, and parsing the data file for problem fields and messages.
- [Setting registry keys for Transaction/Execute](#)
If you are using the Transaction/Execute endpoint in the Datacap Web Services, specifically if the endpoint could be called concurrently by more than one client, you must set registry keys on each Datacap Web Services server.

Example

The following table displays a simplified set of client requests and the return values. This sequence runs an OCR ruleset on the provided image:

Table 1. HTTP requests and returns while running an OCR ruleset on an image

HTT P	URL	Request Body	Result	Return
POS T	/Session/Logon	Application, Password, Station, User	200	Sets cookie
GET	/Transaction/Start	The request for this method contains no content.	201	<i>transactionId</i>
POS T	/Transaction/SetFile/VScan/xml	multipart/form- data content	201	The response for this method contains no content.
POS T	/Transaction/SetFile/tm000001/tif	multipart/form- data content	201	The response for this method contains no content.

HTT P	URL	Request Body	Result	Return
POS T	/Transaction/Execute	Transaction ID, Application, Workflow, PageFile, Rulesets	200	Result set
GET	/Transaction/GetFile/transactionId/_rrs/log	The request for this method contains no content.	200	File contents byte stream
GET	/Transaction/GetFile/transactionId/tm000001/xml	The request for this method contains no content.	200	File contents byte stream
POS T	/Session/Logoff	The request content includes the wTmId cookie the request header.	200	The response for this method contains no content.

In this example, VScan.xml file must be constructed in advance to point to the image, and the parameters that are passed to Transaction/Execute in the request body consist of the *transactionId*, *application*, *workflow*, *PageFile*, and any *rulesets*.

In JSON, the Transaction/Execute request body would be similar to the following:

```
{
  "TransactionId": "2fde5e64-d3ca-4798-824a-feaf880d2099",
  "Application": "MyApp",
  "Workflow": "OCR",
  "PageFile": "VScan.xml",
  "Rulesets": "Batch Profiler"
}
```

Parent topic: [Rule Execution](#)

Setting registry keys for Transaction/Execute

If you are using the Transaction/Execute endpoint in the Datacap Web Services, specifically if the endpoint could be called concurrently by more than one client, you must set registry keys on each Datacap Web Services server.

About this task

If you do not set these registry keys, Datacap Web Services can crash.

Procedure

Set the appropriate registry key on each Datacap Web Services server:
Microsoft Windows (64-bit):

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Datacap\RRS]
"Use_eRRO"=dword:00000001
```

Microsoft Windows (32-bit):

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Datacap\RRS]
"Use_eRRO"=dword:00000001
```

Parent topic: [Running rules directly on images by using Datacap Web Services transactional endpoints](#)

TravelDocs: Stepping a batch through the PageID task profile

Even though you did not add any new functions to the application, you can run the VScan and PageID rulesets again and step through the actions.

Procedure

1. Click the Datacap Studio Test tab.
2. In the Workflow pane, select the VScan task profile under Main Job.
3. Click New to start a new batch.
4. Click Process rules for target object on the main Test tab toolbar.
5. When asked if you want to release the batch, click Advance. The Advance command moves the batch to the next step in the workflow (PageID).
6. Click Step in on the main Test tab toolbar.
7. Click Step in a few times. Execution remains at the batch level. Because the ImageFix Load Settings rule is assigned at the batch level, Datacap expands the rule and prepares to run the LoadSettings action.
8. Continue clicking Step in to run the ImageFix Load Settings action and complete the rule. Because the Enhance Image rule is not assigned at the batch level, it does not run.
9. Continue clicking Step in until the runtime batch hierarchy indicates that first page is selected. Datacap is now ready to run page level rules on each page in the runtime hierarchy, starting with page TM000001.
10. Continue clicking Step in. Because the Enhance Image rule is assigned at the page level, Datacap expands this rule and prepares to run the ImageEnhance action.
11. In the center pane of Datacap Studio, click the Image tab if it is not already active. The Image tab displays the current runtime object.
12. Click Step in to run the ImageEnhance action. The Image pane displays the current page image after image enhancement. The check mark beside the action in the Rulesets pane indicates that the action returned True.
13. Right-click the batch in the Workflow pane and choose Cancel.

Results

For more information, see [Single-stepping through your code](#) and [Using breakpoints](#).

Parent topic: [Rule Execution](#)

Document assembly

Datacap identifies incoming pages and assigns the correct page type by using fingerprint matching or one of the other identification methods. The next step assembles the batch of individual pages into documents according to the rules that are defined within the document hierarchy.

In preparation for the recognition phase, you create a runtime data file for each page.

Document assembly also includes document integrity checking, which confirms that the documents conform to predefined rules. The process also takes corrective action if the documents do not conform.

- [Structured documents](#)
Structured documents are based on the document hierarchy and include data files. These documents conform to predefined integrity rules.
- [TravelDocs: Document creation and page file setup](#)
To create documents and set up the page files, you must create a batch. Then, you can examine the runtime batch folder and review the page data files.
- [TravelDocs: Document integrity management](#)
To manage document integrity issues in the TravelDocs application, you need to configure branching and generate a batch.

Parent topic: [Datacap application development](#)

Structured documents

Structured documents are based on the document hierarchy and include data files. These documents conform to predefined integrity rules.

- [Hierarchy-based documents](#)
The document hierarchy defines both the document types that your application supports and the page types that are associated with each document type.
- [Creation of the page data files](#)
After you arrange individual pages into documents, you must create a runtime data file for each page. The runtime data file is an XML file that includes an element for each field on the page.
- [Document integrity](#)
Even though the document hierarchy defines the required structure of the batch, Datacap might encounter a batch that does not conform to the required structure.
- [Document integrity problem management](#)
You can manage document integrity problems by routing a batch to a job that fixes the problems.

Parent topic: [Document assembly](#)

Hierarchy-based documents

The document hierarchy defines both the document types that your application supports and the page types that are associated with each document type.

For example, the TravelDocs document hierarchy defines three document types, four associated page types, and the generic page type `Other`.

After page identification assigns the correct page type to each incoming page, your application uses information in the document hierarchy to determine the corresponding document. For example, a page of type `Rental_Agreement` is part of a car rental document. A page of type `Air_Ticket` is part of a flight document.

Datacap then uses the information in the document hierarchy to assemble individual pages into multi-page documents. Each page has three variables that define the structure of the parent document:

- *Max*: Maximum number of pages of this type for each document (0 means no maximum).
- *Min*: Minimum number of pages of this type for each document (0 means no minimum).
- *Order*: Position of a page relative to other pages in the same document (0 means any position).

For example, here are the variables that you specified earlier for each of the TravelDocs pages.

	Max	Min	Order	Description
Rental Agreement	1	1	1	One per document; required; must be first
Optional Insurance	1	0	2	One per document; optional; must be second
Air_Ticket	1	1	1	One per document; required; must be first
Room_Receipt	1	1	1	One per document; required; must be first

The variables determine that each car rental document must contain one rental agreement page, and that the page might be followed by an optional insurance page. If Datacap identifies a rental agreement page that is immediately followed by an optional insurance page, it groups the two pages as a single document. The following example is a portion of the runtime data file (PageID.xml) that is generated after document creation.

```
<?xml-stylesheet type="text/xsl"
href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_...\dco.
xsl"?>
<B
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev065_2010
0321.011">
  <V n="TYPE">TravelDocs</V>
  <V n="LAST_RR_TPROFILE">Rulerunner:m:eRun</V>
  <D
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev065_2010
0321.011.01">
  <V n="TYPE">Car_Rental</V>
  <V n="STATUS">0</V>
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev065_TM00
0001">
  <V n="TYPE">Rental_Agreement</V>
  <V n="STATUS">49</V>
  <V n="IMAGEFILE">tm000001.tif</V>
  etc.
  </P>
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev065_TM00
0002">
  <V n="TYPE">Optional_Insurance</V>
  <V n="STATUS">49</V>
  <V n="IMAGEFILE">tm000002.tif</V>
  etc.
  </P>
</D>
etc.
```

- [Assembling documents](#)

The Datacap Studio Actions library includes an action to assemble pages into documents.

Parent topic: [Structured documents](#)

Assembling documents

The Datacap Studio Actions library includes an action to assemble pages into documents.

About this task

Library	Action	Description
DCO	Creates Documents	Assembles the pages of the current batch into documents that are based on the structure that is defined in the document hierarchy and the min, max, and order properties.

Important: You must use this action within a rule that runs at the batch level.

Parent topic: [Hierarchy-based documents](#)

Creation of the page data files

After you arrange individual pages into documents, you must create a runtime data file for each page. The runtime data file is an XML file that includes an element for each field on the page.

The DCO actions library includes the following action to create a runtime data file for the current page.

Library	Action	Description
DCO	Create Fields	Creates a page data (XML) file for the current page. The file includes an element for each field that is defined in the document hierarchy for the current page type. Each field has an identifier and three properties (TYPE, Position, and Status) with default values.

Important: You must run this action within a rule that runs at the page level.

The data file is empty initially, but the CreateFields action uses the document hierarchy structure to create a shell with an element for each data field. The shell gets populated later during recognition.

```
<?xml-stylesheet type="text/xsl"
href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_...\dco.
xsl"?>
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev087_TM00
0001">
    <!-- Page data file for first page in batch (type
Rental_Agreement)
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev087_Pick
up_Date">
    <!-- Pickup_Date field (no data)
        <V n="TYPE">Pickup_Date</V>
        <V n="Position">0,0,0,0</V>
        <V n="STATUS">0</V>
    </F>
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev087_Pick
up_Location">
    <!-- Pickup_Location field (no data)
        <V n="TYPE">Pickup_Location</V>
        <V n="Position">0,0,0,0</V>
        <V n="STATUS">0</V>
    </F>
etc.
```

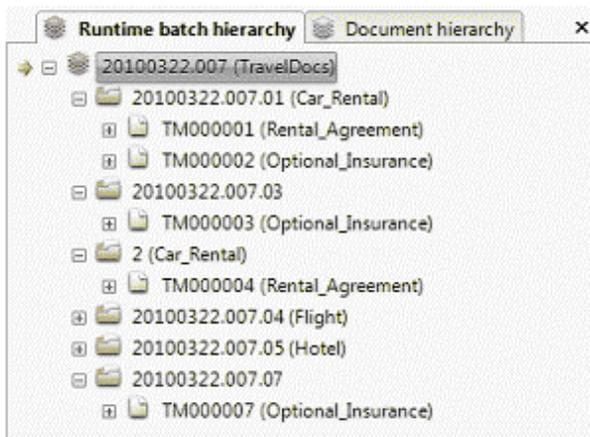
Parent topic: [Structured documents](#)

Document integrity

Even though the document hierarchy defines the required structure of the batch, Datacap might encounter a batch that does not conform to the required structure.

For example, a batch in the TravelDocs application might include a rental agreement page that is followed by two optional insurance pages. Similarly, a batch might include an optional insurance page that is not preceded by a rental agreement page.

The following runtime batch has two structural integrity issues.



- In the first case, the CreateDocuments action grouped the first optional insurance page with the preceding rental agreement page. The action also placed the second insurance page in a separate document of a unidentified type.
- In the second case, the CreateDocuments action again placed the orphaned insurance page in a document of a unidentified type.

In both cases, the batch does not comply with the document integrity rules that are defined in the document hierarchy. However, the CreateDocuments action sets the document status to 0 (OK) and the page status to 49 (ScanOK).

```
<D
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev032_2010
0322.007.03">
  <V n="TYPE"></V>
  <V n="STATUS">0</V>      <-- Document status is "OK" (0)
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev032_TM00
0003">
  <V n="TYPE">Optional_Insurance</V>
  <V n="STATUS">49</V>    <-- Page status is "ScanOK" (49)
  etc.
  </P>
</D>
etc.
<D
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev032_2010
0322.007.07">
  <V n="TYPE"></V>
  <V n="STATUS">0</V>      <-- Document status is "OK" (0)
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev032_TM00
0007">
  <V n="TYPE">Optional_Insurance</V>
```

```
<V n="STATUS">49</V> <-- Page status is "ScanOK" (49)
etc.
</P>
</D>
```

- [CheckAllIntegrity action](#)

The Datacap Studio `rrunner` actions library includes the `CheckAllIntegrity` action that checks the structural integrity of a batch.

Parent topic: [Structured documents](#)

CheckAllIntegrity action

The Datacap Studio `rrunner` actions library includes the `CheckAllIntegrity` action that checks the structural integrity of a batch.

Library	Action	Description
rrunner	CheckAllIntegrity	Returns True if the current batch meets the requirements that are defined in the document hierarchy; returns False otherwise.

Important: You must run this action within a rule that runs at the batch level.

`CheckAllIntegrity` does not change the status variable on non-conforming documents. Instead, the action returns False if there is a document integrity issue, which you can use to trigger corrective action.

In the log file that is generated by the Rulerunner task profile (`rulerunner_rrs.log`), you can see the code that is returned by `CheckAllIntegrity` if the action returns `False`. If the batch includes multiple problems, the code represents the last problem. These codes represent different problems:

- 1 = Has more child objects than allowed by *max* attribute
- 2 = Has fewer child objects than required by *min* attribute
- 3 = A child object is not of a type that is supported by parent
- 4 = A child object is in the wrong position as specified by the *pos* attribute

Parent topic: [Document integrity](#)

Document integrity problem management

You can manage document integrity problems by routing a batch to a job that fixes the problems.

The Document Integrity ruleset demonstrates how to handle document integrity problems. The `Batch Route To Fixup` function is started only if `CheckAllIntegrity` returns a value of `False`.

A document integrity problem can be pages in the wrong order, or a missing required page. If `CheckAllIntegrity` identifies a document integrity issue, the application sends the batch to a `Fixup` job. An operator then fixes the problem and returns the batch to the main workflow. Moving a batch out of the current workflow in this way is known as branching.

Note: The Datacap Studio application wizard generates the Profiler task profile that includes recognition and validation. Therefore, Datacap completes recognition and validation before the batch branches to the `Fixup` job. So, some pages might have `Status = 1`, which indicates a problem. You cannot complete the `Fixup` job unless the `Status` on all pages is 0, which indicates that there is no problem. You can configure your application to use two separate task profiles so that Datacap branches to `FixUp` before recognition if there are document integrity problems. For details about using two task profiles, see the topic [Creating the CreateDocs task](#).

The `Batch Route To Fixup` function uses these `runner` actions to branch to the Fixup job.

Library	Action	Description
runner	Task_NumberOfSplits	Specifies the number of jobs to which the batch is sent to before the branch returns to the main workflow (almost always 1).
runner	Task_RaiseCondition	Specifies the group index (almost always 0) and the index of the condition to raise, where 0 is the first condition. The index is including on a list on the Datacap Web Client, Administrator tab, Workflow page. To view the Workflow page, start the Datacap Web Client, log on to the TravelDocs application, click the Administrator tab, and click Workflow.

Before Datacap can branch to the FixUp job, you must configure the settings on the `Document Integrity Failed` condition. See the topic [Configuring branching](#).

Parent topic: [Structured documents](#)

TravelDocs: Document creation and page file setup

To create documents and set up the page files, you must create a batch. Then, you can examine the runtime batch folder and review the page data files.

- [Running a batch through the workflow](#)
Because no functions are added to the CreateDocs ruleset yet, you can run the default version that is generated by the Datacap Studio application wizard.
- [Contents of the runtime batch folder](#)
To examine the runtime batch folder, open the most recent batch folder (C:\Datacap\TravelDocs\batches*batch_identifier*).
- [Page data files](#)
The `CreateFields` action in the `Create Fields` rule creates a page data file for the current page.

Parent topic: [Document assembly](#)

Running a batch through the workflow

Because no functions are added to the CreateDocs ruleset yet, you can run the default version that is generated by the Datacap Studio application wizard.

Procedure

1. Click the Datacap Studio Test tab.
2. In the Workflow pane, select the VScan task profile under Main Job.
3. Click New to start a new batch.
4. Click Process rules for target object on the main Test tab toolbar.

5. When you are prompted to release the batch, click Advance. The Advance command moves the batch to the next step in the workflow (PageID).
6. Click Process rules for target object on the main Test tab toolbar.
7. When you are prompted to release the batch, click Advance. The Advance command moves the batch to the next step in the workflow (Profiler).
8. Click Process rules for target object on the main Test tab toolbar.
9. When you are prompted to release the batch, click Advance. The Advance command moves the batch to the next step in the workflow (Verify).
10. On the Runtime batch hierarchy tab, expand each document node to see how the individual pages are now grouped into documents. The flight document and the hotel document all have only one page, but two of the car rental documents have multiple pages.
11. Right-click the running batch icon in the Workflow pane and choose Cancel.

Parent topic: [TravelDocs: Document creation and page file setup](#)

Contents of the runtime batch folder

To examine the runtime batch folder, open the most recent batch folder (C:\Datacap\TravelDocs\batches*batch_identifier*).

The runtime batch folder contains the following files:

File	Description
TM00000*.tif	An image-enhanced version of each of the sample image files.
TM00000*.tio	A copy of each of the original image files.
TM00000*.xml	The page data file for each image file.
TM00000*.c.xml	The results of full page recognition for each image file.
TM00000*.cco	The fingerprint file for each of the image files.
Profiler.xml	The runtime document hierarchy that is generated by the Profiler task profile.
Profiler_rrs.log	The log file that is generated by the Profiler task profile
PageID.xml	The runtime document hierarchy that is generated by the PageID task profile.
pageid_rrs.log	The log file that is generated by the PageID task profile.
VScan.xml	The runtime document hierarchy that is generated by the VScan task profile.
vscan_rrs.log	The log file that is generated by the VScan task profile.
Verify.xml	A copy of the runtime document hierarchy ready for use by the next task profile in the workflow (Verify).

Parent topic: [TravelDocs: Document creation and page file setup](#)

Page data files

The `CreateFields` action in the `Create Fields` rule creates a page data file for the current page.

The data file includes all of the fields that are identified for the current page type in the document hierarchy. Each field has an identifier (ID) and three properties: `TYPE`, `Position`, and `Status`.

```
<?xml-stylesheet type="text/xsl"
href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_...\dco.
xsl"?>
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev228_TM00
0001">      <!--Page data file for first page in batch (type Rental_Agreement)
      <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev228_Pick
up_Date">      <!--Pickup_Date field (no data)
      <V n="TYPE">Pickup_Date</V>
      <V n="Position">0,0,0,0</V>
      <V n="STATUS">0</V>
      </F>
      <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev228_Pick
up_Location"> <!--Pickup_Location field (no data)
      <V n="TYPE">Pickup_Location</V>
      <V n="Position">0,0,0,0</V>
      <V n="STATUS">0</V>
      </F>
      etc.
```

Later, other actions can assign different values to these properties and add properties as needed.

Parent topic: [TravelDocs: Document creation and page file setup](#)

TravelDocs: Document integrity management

To manage document integrity issues in the TravelDocs application, you need to configure branching and generate a batch.

- [Configuring branching](#)
The Datacap Studio application wizard generates `Document Integrity` ruleset that is configured to identify document integrity problems and send the batch to the Datacap Desktop Fixup task. However, more setup steps are required in the Datacap Web Client Administrator tab to configure the required branching.
- [Running a batch with document integrity problems](#)
For demonstration purposes, an optional insurance page is added to the end of the batch. Because the page does not immediately follow a car rental agreement page, the `CheckAllIntegrity` action generates an error during document integrity checking.

Parent topic: [Document assembly](#)

Configuring branching

The Datacap Studio application wizard generates `Document Integrity` ruleset that is configured to identify document integrity problems and send the batch to the Datacap Desktop Fixup task. However, more setup steps are required in the Datacap Web Client Administrator tab to configure the required branching.

Procedure

To configure branching in the TravelDocs application:

1. Open the Administrator tab in the Datacap Web Client for TravelDocs and click Workflow.
2. Expand Main job and then expand Profiler.
3. Select the Document Integrity Failed condition and, if necessary, configure the values as follows:
 - o Spawn type: Branch
 - o Child Job: Fixup Job
 - o Parent Status: Pending
 - o Child status: Pending
 - o Steps: 1
4. Click Save Condition (if you had to change the default values). The application is configured to branch to the Fixup job when document integrity fails, and then return to the main job with a status of `pending`.
5. Leave the Datacap Web Client open.

Parent topic: [TravelDocs: Document integrity management](#)

Running a batch with document integrity problems

For demonstration purposes, an optional insurance page is added to the end of the batch. Because the page does not immediately follow a car rental agreement page, the CheckAllIntegrity action generates an error during document integrity checking.

Procedure

1. Open C:\Datacap\TravelDocs\images.
2. Make a copy of Images_Page_02.tif and name the copy Images_Page_12.tif. Making a copy creates an orphaned insurance page at the end of the batch.
3. In the Datacap Web Client Operations tab, click VScan.
4. When the task is completed, click Stop.
5. On the Operations tab, click Upload.
6. When the Upload task displays a message to indicate that the task is complete, click OK and then click Stop.
7. Start the Datacap Desktop application. In the Start menu, select IBM Datacap Clients > Datacap Desktop. Log on to the TravelDocs application with the `Admin` account, and select PageID from the Shortcut menu.
8. Select Profiler from the Shortcut menu. When the task completes, click Stop.
9. In Datacap Web Client, select the Monitor tab to display the Job Monitor page. Notice that the batch has two entries in the queue.
 - o The first entry indicates that the batch has a status of `Pending` for the Fixup task.
 - o The second entry indicates that the batch has status of `Waiting` for the Verify task.
10. In Datacap Desktop, log on to the TravelDocs application with the `Admin` account, and select Fixup from the Shortcut menu. The batch opens in the Datacap Desktop Fixup window. The last document is selected and the Comments field indicates that the document has an invalid member (the orphaned insurance page).
11. Select the page `TM000012`, click Delete, and click OK to confirm.
12. Click Finish and then click OK in the Task Finished message.
13. In the Datacap Web Client Job Monitor page, press the F5 key to refresh the view. The Fixup task now has a status of `Job done` and the batch is now pending for the Verify task.
14. Open the C:\Datacap\TravelDocs\images folder and delete the file Images_Page_12.tif to avoid a recurrence of this problem.

Parent topic: [TravelDocs: Document integrity management](#)

Data recognition

Data recognition is the stage during which you locate the fields that you want to capture and then convert the fields into character-based data.

The data that is obtained from recognition is stored in the page data files that you set up in the document assembly stage.

There are several techniques that you can use to identify pages. The most widely used is fingerprint matching. If you used fingerprint matching for page identification, you most likely used the fingerprint images to define the recognition zones. These zones are the fields that you want to read on each page. If you use full page recognition, you can obtain the field data directly from the full page recognition results. Otherwise, you need to run the recognition engine on each field zone to capture the data.

The other recognition techniques do not use fingerprint zones to locate the field data. Instead, they use text matching or pattern matching to analyze the page and identify the fields.

- [Page data recognition](#)
The recognition of page data includes the using fingerprints to identify recognition zones, storing the recognition zone information, and reading data from the page.
- [Dynamic locale support](#)
The Datacap Recognition Engines use locale variables to validate language and regional settings such as currency, numbers, and date data types. You can use dynamic locale support to set different locale variables on the nodes of your application that use different languages.
- [Check box options management](#)
Managing check box options requires that you establish the parent fields and their required variables, and then use either OCR/A recognition or pixel threshold evaluation.
- [Identifying medical claim forms by using Autofield](#)
Datacap Medical Claims uses Autofield in addition to fingerprint matching to recognize medical claim pages and fields.
- [TravelDocs: Specification of recognition zones](#)
You must define the positions of the various fields for one variant of each page type. As Datacap Studio locates field zones that are on the fingerprint images, it writes the position information for each field into the document hierarchy.
- [TravelDocs: Assignment of default rules to the document hierarchy](#)
When the Datacap Studio application wizard generates the application framework, it attaches only default rules to default elements. Therefore, you must attach rules to the new elements that you created.
- [TravelDocs: Updating the application to manage check box options](#)
You need to configure required variables, specify the check mark type, and create a rule to recognize the OMR fields to manage check box options.
- [TravelDocs: Using pixel threshold check box recognition \(optional\)](#)
For demonstration purposes, you can go through the pixel threshold method.

Parent topic: [Datacap application development](#)

Page data recognition

The recognition of page data includes the using fingerprints to identify recognition zones, storing the recognition zone information, and reading data from the page.

- [Identifying recognition zones by using fingerprints](#)
Datacap uses fingerprints to identify incoming pages. For each incoming page, Datacap generates a

fingerprint file that describes the page, and compares the new fingerprint to a fingerprint library for known page types. When Datacap finds a match, it assigns the corresponding page type

- [Recognition zone information storage](#)

Datacap Studio stores the coordinates for each field zone as a variable in the document hierarchy.

- [Reading data from the page](#)

The method that you use to read data from a page depends on the method that you used to generate the runtime fingerprints.

Parent topic: [Data recognition](#)

Identifying recognition zones by using fingerprints

Datacap uses fingerprints to identify incoming pages. For each incoming page, Datacap generates a fingerprint file that describes the page, and compares the new fingerprint to a fingerprint library for known page types. When Datacap finds a match, it assigns the corresponding page type

The fingerprint library has a second purpose, which is to identify the position of each field for each known page type. There can be many variants of each page type and the position of each field is different for each variant. So you must identify the recognition zones for each variant.

Parent topic: [Page data recognition](#)

Recognition zone information storage

Datacap Studio stores the coordinates for each field zone as a variable in the document hierarchy.

The pickup date field on the Car Rental #1 page and Car Rental #2 page has the following zones:

- The Car Rental #1 fingerprint has ID 695, so Pos695 defines the position of the pickup date field for Car Rental #1.
- The Car Rental #2 fingerprint has ID 678, so Pos678 defines the position of the pickup date field for Car Rental #2.

You can see the position coordinates for a field by right-clicking the field name in the Document Hierarchy pane. Choose Manage variables (the document hierarchy must be locked). Alternatively, you can select the field and look in the Properties pane.

For every page that Datacap identifies a page as a Car Rental #2 rental agreement, the pickup date is at coordinates (579,353,943,415).

Important: The coordinates are relative to a reference point on the page. If the incoming page does not align precisely with the reference page, Datacap calculates an offset and uses it to determine the actual field positions.

Parent topic: [Page data recognition](#)

Reading data from the page

The method that you use to read data from a page depends on the method that you used to generate the runtime fingerprints.

- If you used full page recognition to generate the runtime fingerprints, you can obtain the field data directly from the fingerprint (CCO) file.
- If you used AnalyzeImage to generate the runtime fingerprints, you must use recognition on each of the field zones to obtain the field data.

Populating the page data files using full page recognition results

The Datacap Studio Actions library include actions that take the character data from the fingerprint (CCO) file and apply it to the runtime batch hierarchy.

Library	Action	Description
Zones	ReadZones	Loads the zone position information for the current fingerprint.
Recog_Shared	SnapCCOtoDCO	Transfers the recognition results from the current page's fingerprint (CCO) file to the appropriate field objects in the runtime batch hierarchy.

You must run ReadZones before you can run SnapCCOtoDCO.

Populating the page data files using zone OCR

If you used AnalyzeImage to generate the runtime fingerprints, the following actions are available for zone-based recognition:

Library	Action	Description
Zones	ReadZones	Loads the zone position information for the current fingerprint.
ocr_a	RecognizePageFieldsOCR_A	Recognizes the characters within each of the field zones using the position information in the current fingerprint.
OCR_s	RecognizePageFieldsOCR_S	Recognizes the characters within each of the field zones using the position information in the current fingerprint.
ocr_sr	RecognizePageFieldsOCR_S	Recognizes the characters within each of the field zones using the position information in the current fingerprint.
icr_c	RecognizePageFieldsICR_C	Recognizes the characters within each of the field zones using the position information in the current fingerprint.

Parent topic: [Page data recognition](#)

Dynamic locale support

The Datacap Recognition Engines use locale variables to validate language and regional settings such as currency, numbers, and date data types. You can use dynamic locale support to set different locale variables on the nodes of your application that use different languages.

The value of the locale variable is inherited by to all of the child nodes of the batch object. Unless a different locale value is assigned to a child node. Then, all of the child nodes of that object inherit the new locale value. For example, if you add a locale value to a Document, that locale value is also assigned to the Pages and fields in that Document. If you then set a different locale value to a Page in the Document, that new value is also assigned to the fields in that Page.

You can set the locale variable for document processing by the application at the system level, the application level, or the node level in the application.

By default, the Windows Regional and Language locale is used system wide for all validations that are done during recognition. Unless the locale variable is set differently elsewhere.

For applications that use different languages, you can set the locale variable at the application level in the Datacap Application Manager. This setting is used on the batch object and is inherited down to of the all child

nodes in the batch object. Setting the locale variable on the application level overrides the Windows Regional and Language setting.

If your applications run workflows, documents, pages, or fields that use different languages, you can set the locale variable at any node level in the DCO. Setting the locale variable on these levels overrides the locale value in Datacap Application Manager.

All of the standard actions, including the Recognition Engine, use the Locale property setting for the node to which the action is bound. The action property setting dynamically overrides the locale values that are set at the system, application, or batch object levels. For example, if the `rrSet(en-US,@P.hr_locale)` action is bound on the Page node. That node and its child nodes use the setting for English US (en-US) locale. Regardless of the locale that is set on the application level or in the Batch or Document level of the DCO.

- [Setting locale values](#)
You set locale values on an application or batch object to validate the language and regional settings for that node. You can dynamically override the locale values that are inherited by the child nodes of the batch object. To override these locale values, assign a different locale value to the child node.
- [Recognition language settings](#)
You can process documents in different languages. You specify the language that is used by the recognition engines by setting the recognition language value on the node. You can set language values to recognize different languages on the documents, pages, and fields of your application.
- [Enabling automatic language detection for OCR_A recognition](#)
You can specify a list of expected languages that the OCR_A Recognize action will need to detect. After you set this list, the recognition engine detects the language automatically while OCR runs in the Recognize action.
- [Supported language codes](#)
Datacap provides language support for many countries and regions around the world. You use language codes to set the locale that is associated with the language and regional settings on the documents that are processed by your application.

Parent topic: [Data recognition](#)

Setting locale values

You set locale values on an application or batch object to validate the language and regional settings for that node. You can dynamically override the locale values that are inherited by the child nodes of the batch object. To override these locale values, assign a different locale value to the child node.

About this task

You can set locale variables at the following levels:

System-wide locale settings

Use the Windows Regional and Language settings.

Applications that use one language or multiple languages

Set the locale value at the application level in the Datacap Application Manager.

Applications with workflows, documents, pages, or fields that use different languages

Set the locale value at any node level in the DCO or on an action that is bound to a node.

Procedure

To set the locale on an application or on the nodes in an application:

1. Assign a locale value to an application.
 - a. In the Start menu select IBM Datacap Services Datacap Application Manager.
 - b. In the Applications pane, select the application for which you want to set the locale.
 - c. Click the Main tab.
 - d. In the Locale field, select the language that you want and press Enter. You can select the System option from the Locale language list to reset the locale value. When you set the locale in the Datacap Application Manager, at run time the specified value is automatically created. The value is created in the `hr_locale` variable at the batch level of the runtime DCO.
 - e. Run the following steps. Set the locale property on the action to change this value at run time when you must change the current language during task execution.
2. Set the locale value on nodes in the DCO:
 - a. In Datacap Studio, click the Document Hierarchy tab.
 - b. Click the Lock DCO for Editing icon to lock the application for editing.
 - c. Select the node to which you want to set the locale.
 - d. Right-click and select Manage Variables and then click New.
 - e. Type `hr_locale` and press Enter.
 - f. Type the language code for the `hr_locale`.
 - g. Click Done, then click Save Changes.
 - h. Select the node and click the Properties tab to verify that `hr_locale` was added.
 - i. Click Unlock DCO.
3. Set the locale property on the action:
 - a. In Datacap Studio, click the Rulemanager tab.
 - b. Select the ruleset for which you want to set the locale.
 - c. Click Lock ruleset for Editing. Locking the ruleset does not lock the DCO.
 - d. Add the `rrSet` action to the ruleset.

Important: The `rrSet` action must precede the action on the node to which you want to set the locale property.
 - e. Using smart parameter syntax, configure the `rrSet` action to set the `hr_locale` variable to the locale value that you want. For example, use `rrSet(en-US,@P.hr_locale)` on a Page object or `rrSet(en-US,@F.hr_locale)` on a field object. The locale setting is used on all of the subsequent actions in the ruleset.
 - f. Click Save to save the changes to the ruleset. The ruleset is published and unlocked.

Parent topic: [Dynamic locale support](#)

Recognition language settings

You can process documents in different languages. You specify the language that is used by the recognition engines by setting the recognition language value on the node. You can set language values to recognize different languages on the documents, pages, and fields of your application.

A recognition engine can support the recognition of multiple languages. For these engines, you can set the language that is recognized for the entire page or for each field on the page. Setting recognition values on the Document and Page levels can recognize multiple language types within a single application. For example, you can configure Page A to be recognized in English and Page B to be recognized in French. Setting locale values at the field level enables multiple language recognition on a single page.

In addition to the locale specification, specific engine settings to configure the recognized language are also available by using Datacap Studio. On the Zones tab, you use the appropriate recognition engine tab, such as the OCR/S tab for the OCR/S engine. You can set the language settings in the Language Environment, Recognition Setup, and Spell Check sections of the Properties tab. If you set these values here, the recognition engine uses these values first. If these settings are not set here, the recognition engine uses the locale setting

that is configured by the Datacap Application Manager. In this case, the Datacap Application controls the `hr_locale` variable.

The language value on the Zones > Properties tab for the recognition engine is English, even if a value was not set. To confirm that the language value was set to English, you can check the following DCO variables for the recognition engine:

- `OCR/S: s_lg`,
- `OCR/A: y_lg`
- `ICR/C: c_cr`.

Important: If the `c_cr` variable for `ICR_C` recognition is not set for the engine, you can set it manually. In Datacap Studio, set the `c_cr` variable to `USA` prior to the call to recognition as follows:

```
SaveAsCurrentObjVariable("c_cr", "USA")
RecognizeFieldICR_C
```

The Properties tab of the recognition engine also contains a Use Locale setting. If the Use Locale value is set to Yes, the recognition engine must use the value of the `hr_locale` variable. Even if the language settings for the engine are set. If you are using OCR/S and are recognizing simplified Chinese, you must set the OCR/S Module setting to Asian recognition. When you use the Asian recognition module, the Filter setting is not used.

Parent topic: [Dynamic locale support](#)

Enabling automatic language detection for OCR_A recognition

You can specify a list of expected languages that the OCR_A Recognize action will need to detect. After you set this list, the recognition engine detects the language automatically while OCR runs in the Recognize action.

About this task

Automatic language detection is enabled by setting the `y_lg` variable on the page and specifying at least three languages in the variable. If the variable is not set, then the language that is recognized is based on the current locale.

Tip: Minimize the list of languages to only the languages that are expected to be processed by the application. The more languages are specified, the slower the processing.

Although the Fine Reader engine supports recognition of a vast number of languages, the language auto-detection feature only works on those languages for which is implemented full dictionary support. These include most popular languages, but does not include Simplified Chinese or Traditional Chinese.

Procedure

1. Use the `rrSet` action or a similar action to set the `y_lg` variable.
2. Set the `y_lg` variable to a comma separated list of at least three languages from the following list:
Important: When setting the comma separated list of languages, be sure that the languages are spelled as written below. An invalid language name will cause the action to abort.

Languages supported in automatic language detection

Remember: This is a list of language names that set the scope of automatic language detection in the OCR engine. This is not a list of languages supported by IBM® Datacap. For information on language support, search the [IBM Support Portal](#) for the language support techdoc applicable to your version of Datacap.

- `ArmenianEastern`

- ArmenianGrabar
- ArmenianWestern
- AzeriLatin
- Bashkir
- Bulgarian
- Catalan
- Croatian
- Czech
- Danish
- Dutch
- English
- Estonian
- Finnish
- French
- German
- GermanNewSpelling
- Greek
- Hungarian
- Indonesian
- Italian
- Japanese
- Korean
- KoreanHangul
- Latin
- Latvian
- Lithuanian
- Norwegian
- NorwegianBokmal
- NorwegianNynorsk
- OldEnglish
- OldFrench
- OldGerman
- OldItalian
- OldSpanish
- Polish
- PortugueseBrazilian
- PortugueseStandard
- Romanian
- RussianOldSpelling
- Russian
- RussianWithAccent
- Slovak
- Slovenian
- Spanish
- Swedish
- Tatar
- Turkish
- Ukrainian

3. Call the Recognize action.

Example

```
rrSet ("English, French, Japanese", "@P.y_lg")
Recognize ()
CreateCcoFromLayout ()
```

This sequence creates a layout XML file and subsequently a CCO file for the current page. Auto detection is enabled for English, French, and Japanese documents. The CCO file that is produced is ready for use by navigation and pattern matching actions.

Parent topic: [Dynamic locale support](#)

Related information:

[CreateCcoFromLayout](#)

Supported language codes

Datacap provides language support for many countries and regions around the world. You use language codes to set the locale that is associated with the language and regional settings on the documents that are processed by your application.

The Recognition Engine actions use the locale property to assign locales to the node to which the action is bound if the engine-specific recognition language settings are not set. For example, if the `rrSet(en-US,@D.hr_locale)` action is bound on a Document node, that node and its child nodes use the English US (en-US) locale. Regardless of the locale setting on the application level or in the DCO. Recognition engines do not necessarily support all of the languages that are specified in the following language tables.

Use the language codes in the following tables to set the locale on Datacap actions.

Eastern European and Russian language codes

Table 1. Supported Eastern European and Russian languages by country

Language	Code
Czech (Czech Republic)	cs-CZ
Croatian (Latin, Bosnia, and Herzegovina)	hr-BA
Croatian (Croatia)	hr-HR
Hungarian (Hungary)	hr-HU
Polish (Poland)	pl-PL
Romanian (Romania)	ro-RO
Russian (Russia)	ru-RU
Slovak (Slovakia)	sk-SK
Turkish (Turkey)	tr-TR

English language codes

Table 2. Supported English languages by country

Language	Code
English (Caribbean)	en-029

Language	Code
English (Australia)	en-AU
English (Belize)	en-BZ
English (Canada)	en-CA
English (Ireland)	en-IE
English (India)	en-IN
English (Jamaica)	en-JM
English (Malaysia)	en-MY
English (New Zealand)	en-NZ
English (Republic of the Philippines)	en-PH
English (United Kingdom)	en-UK
English (United States)	en-US
English (Zimbabwe)	en-ZW

French language codes

Table 3. Supported French languages by country

Language	Code
French (Belgium)	fr-BE
French (Canada)	fr-CA
French (Switzerland)	fr-CH
French (France)	fr-FR
French (Luxembourg)	fr-LU
French (Monaco)	fr-MC

German language codes

Table 4. Supported German languages by country

Language	Code
German (Austria)	de-AT
German (Switzerland)	de-CH
German (Germany)	de-DE
German (Liechtenstein)	de-LI
German (Luxembourg)	de-LU

Spanish language codes

Table 5. Supported Spanish languages by country

Language	Code
Spanish (Argentina)	es-AR
Spanish (Bolivia)	es-BO
Spanish (Chile)	es-CL
Spanish (Columbia)	es-CO
Spanish (Costa Rica)	es-CR
Spanish (Dominican Republic)	es-DO
Spanish (Ecuador)	es-EC
Spanish (Spain)	es-ES
Spanish (Guatemala)	es-GT
Spanish (Honduras)	es-HN
Spanish (Mexico)	es-MX
Spanish (Nicaragua)	es-NI
Spanish (Panama)	es-PA
Spanish (Peru)	es-PE
Spanish (Puerto Rico)	es-PR
Spanish (Paraguay)	es-PY
Spanish (El Salvador)	es-SV
Spanish (United States)	es-US
Spanish (Uruguay)	es-UY
Spanish (Venezuela)	es-VE

Other language codes

Table 6. Other supported languages by country

Language	Language code
Chinese (simplified)	zh-Hans
Dutch (Belgium)	nl-BE
Dutch (Netherlands)	nl-NL
Italian (Italy)	it-IT
Italian (Switzerland)	it-CH
Portuguese (Brazil)	pt-BR
Portuguese (Portugal)	pt-T
Swedish (Finland)	sv-FI

Language	Language code
Swedish (Sweden)	sv-SE

For detailed information about Datacap language support, see the Datacap Language Support techdoc at <http://www.ibm.com/support/docview.wss?uid=swg27047095>

Parent topic: [Dynamic locale support](#)

Check box options management

Managing check box options requires that you establish the parent fields and their required variables, and then use either OCR/A recognition or pixel threshold evaluation.

- [Check box recognition methods](#)
Datacap employs optical mark recognition (OMR) to determine whether a check box option is selected.
- [Establishing parent fields](#)
When you process pages with check box options, you must define the check box options as subfields of a parent field. You must also outline the subfields and the parent field when you are drawing the recognition zones.
- [Setting the required variables on the parent field](#)
To process check box options as OMR fields, you must set the `RecogType` variable on the parent field equal to 4. This setting instructs the Datacap recognition engine to use mark recognition rather than character recognition.
- [Implementing the OCR/A check box recognition method](#)
The OCR/A check box recognition method uses the `RecognizeFieldOCR_A` action to determine whether the zone represents a selected check box or a non-selected check box. You must include this action in a rule that is bound to the parent zone. You must also configure the OCR/A settings for the parent zone (not the individual check box subfields).
- [Using the pixel threshold evaluation method](#)
The pixel threshold evaluation method uses the `RecogOMRThreshold` action in the `Recog_Shared` library.

Parent topic: [Data recognition](#)

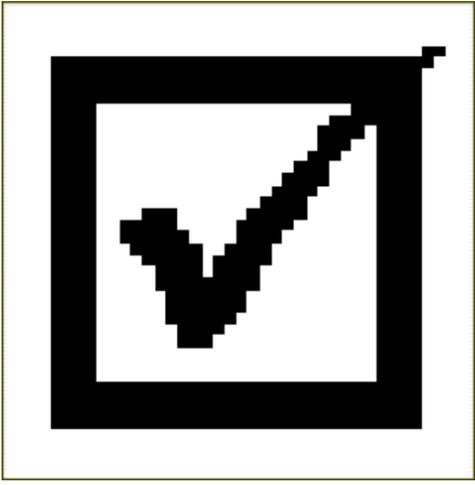
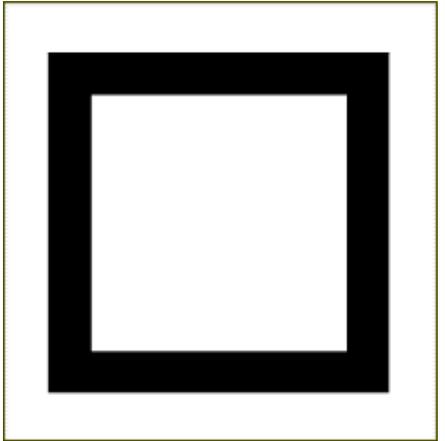
Check box recognition methods

Datacap employs optical mark recognition (OMR) to determine whether a check box option is selected.

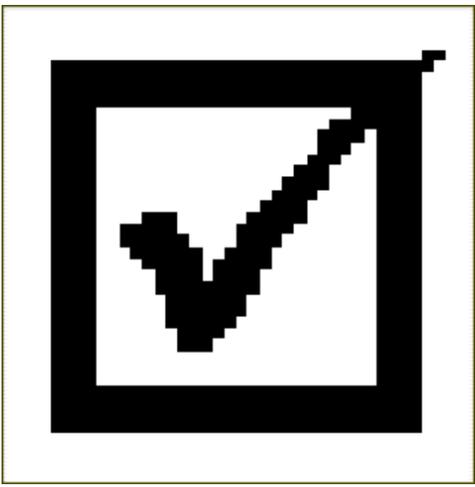
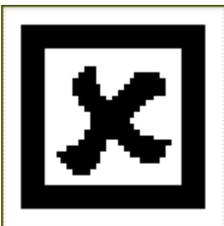
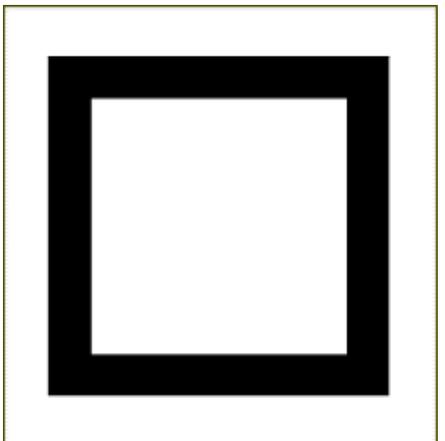
There are two basic OMR techniques.

- **OCR/A check box recognition method:** This method is easy to set up and works well with non-dropout check boxes (where the check box outline remains on the page image). The method does not work as well with drop-out check box (where the outline drops out during scanning). The OCR/A recognition engine determines whether the specified region represents a selected check box (1) or a non-selected check box (0).

Selected	Selected	Not selected
-----------------	-----------------	---------------------

Selected	Selected	Not selected
		

- Pixel threshold evaluation method: This method is more difficult to set up but is more reliable for drop-out check boxes. The method can also be used to read filled-in *bubbles* (O) on a response form. It calculates the percentage of black pixels within a specified zone and compares the result to a predetermined threshold value. For example, if the threshold is 20%, any OMR zone with more than 20% black pixels is considered selected (1). Any zone with 20% or less is considered not selected (0).

> 20% black	> 20% black	<= 20% black
		

The field setup requirements are the same for both setup check box recognition techniques.

Parent topic: [Check box options management](#)

Establishing parent fields

When you process pages with check box options, you must define the check box options as subfields of a parent field. You must also outline the subfields and the parent field when you are drawing the recognition zones.

You defined subfields for the options on the rental agreement page in the TravelDocs application. When you define the recognition zones, you need to define the positions of the parent fields and the subfields.

It might seem easier to draw the child field zone within the check box outline. However, recognition succeeds only if the check boxes on the runtime page align perfectly with the fingerprint image. Even a slight

misalignment can result in a false positive. The best approach is to draw the recognition zone around the check box.

You also defined parent fields and subfields on the optional insurance page, although in this case you created a parent field for each check box. When you define the zones, you need a zone for each parent and each subfield.

Parent topic: [Check box options management](#)

Setting the required variables on the parent field

To process check box options as OMR fields, you must set the `RecogType` variable on the parent field equal to 4. This setting instructs the Datacap recognition engine to use mark recognition rather than character recognition.

If the business requirements indicate that multiple options within the group can be selected, you must set the `MultiPunch` variable on the parent field to 1.

The `RecogType` and `MultiPunch` variables are not included by default, so you must add them manually to the parent object, as described later for the TravelDocs application see [Setting the required variables on the Options and Insurance fields](#).

Parent topic: [Check box options management](#)

Implementing the OCR/A check box recognition method

The OCR/A check box recognition method uses the `RecognizeFieldOCR_A` action to determine whether the zone represents a selected check box or a non-selected check box. You must include this action in a rule that is bound to the parent zone. You must also configure the OCR/A settings for the parent zone (not the individual check box subfields).

You configure the OCR/A settings for specific zones by using the OCR/A tab in the Properties pane on the Datacap Studio Zones tab. The OCR/A tab is not displayed by default, but you can activate it by right-clicking any existing tab, choosing Show tabs, and selecting OCR/A.

Click the OCR/A tab to access the settings that the OCR/A recognition engine uses when it completes recognition on the selected field.

There are three OMR settings:

Table 1. OMR settings

Setting	Description
Check mark type	Select Square background to read non-dropout check box. This setting is stored in the document hierarchy by using the <code>OMRType</code> variable, where 0 is "Square background". <V n="OMRType">0</V>
Length	The Length setting reflects the number of OMR subfields and is set automatically.
Multipunch	The Multipunch setting is the same as the <code>MultiPunch</code> variable you looked at earlier, where 1 is Yes. <V n="MultiPunch">1</V>

Parent topic: [Check box options management](#)

Using the pixel threshold evaluation method

The pixel threshold evaluation method uses the `RecogOMRThreshold` action in the `Recog_Shared` library.

Specifying the threshold and background levels

The `RecogOMRThreshold` action takes two parameters:

- **Threshold:** Specifies the percentage of black pixels over which the option is considered selected.
- **Background:** Used to determine the confidence level and specifies the percentage that can be attributed to the check box outline plus any scanner noise:
 - Any zone with a percentage of black pixels below this value is considered not selected with high confidence. Any zone with a percentage of black pixels between this value and the threshold value is considered not selected with low confidence.
 - Any zone with a percentage of black pixels over $(2 * \text{Threshold} - \text{Background})$ is considered selected with high confidence. Any zone with a percentage of black pixels between Threshold and $(2 * \text{Threshold} - \text{Background})$ is considered selected with low confidence.

However, if `MultiPunch=0` (or is not specified) then only the zone with the highest percentage is selected.

For example, if the threshold value is 20 and the background value is 15 then the high confidence threshold is $(2 * 20 - 15) = 25$. If you run `RecogOMRThreshold (20,15)` on an OMR group field with `MultiPunch=1`, then the following is true.

- Any zone with more than 25% black pixels is considered selected with high confidence
- Any zone with between 20% and 25% is considered selected with low confidence
- Any zone with between 15% and 20% is considered not selected with low confidence
- Any zone with 15% or less black pixels is considered not selected with high confidence

Determining the appropriate threshold and background values

To determine appropriate values for the threshold and background parameters, you must determine the percentage of pixels within the OMR zone that can be attributed to the check box outline plus any scanner noise. The easiest way to make this determination is to run a page that contains both checked and cleared option boxes through the workflow. Then get the pixel counts from the page data file.

When `Datacap` runs a `RecogOMRThreshold` action, it counts the number of black pixels within each OMR zone. `Datacap` then writes the resulting values to the page data file as a density string.

```
</F>
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev384_Options">
  <V n"Type">Options</V>
  <V n"Position">1171,327,1518,622</V>
  <V n"STATUS">0</V>
  <V n"DensityString">FBG</V>
  <C cn="10" cr="1440,405,1490,418">49</C>
  <C cn="10" cr="1440,475,1490,525">48</C>
  <C cn="10" cr="1440,541,1490,591">49</C>
</F>
```

The `DensityString` has one character per OMR zone. In this code example, the `Options` field has three OMR zones and the `DensityString` value is `FBG`. Each of the characters corresponds to a percentage value according to the following formula.

Percentage black pixels = character's ASCII code value minus 48.

In this example:

- The ASCII code for each of the three characters is 70, 66, and 71 respectively.
- The percentage of black pixels for each of the three zones is 22% (70-48), 18% (66-48), and 23% (71-48), respectively.

After you obtain the percentage values, you can then refer to the original page image to see if the corresponding check box is selected. This example was obtained from a page where the first and third options were selected, and the second option was not selected.

Checkbox	Percentage filled
Check mark	22%
Empty Square	18%
Check mark	23%

Based on these three check boxes, you set the threshold and background values somewhere between 18 and 22. (Fractional values are permitted for the threshold and background parameters.) You can test the values scanning additional pages and checking their density strings before setting final values.

Implications of using `RecogOMRThreshold`

The `RecogOMRThreshold` action relies upon pixel counts within the OMR zone. So it is important that all OMR zones have the same dimensions, or as close as possible.

Drawing OMR zones on the Datacap Studio Zones tab can sometimes be difficult. You can establish approximate zone boundaries by drawing the bounding boxes on the Image View tab. Then edit the coordinates in the *Pos* variables field in the Properties pane.

The coordinates correspond to the upper left corner of the bounding box, such as the *x1*, *y1* coordinate, and the lower right corner (*x2*, *y2*). In this example, you enter *x1*, *y1*, *x2*, *y2* in the *Pos* variable field.

Parent topic: [Check box options management](#)

Identifying medical claim forms by using Autofield

Datacap Medical Claims uses Autofield in addition to fingerprint matching to recognize medical claim pages and fields.

While Datacap compares fingerprint images to a library of fingerprints for known page types, Autofield works by fitting pixels in an image to a particular zone. As a consequence, changing a zone by moving it by just a few pixels makes scanned medical claims unrecognizable to Medical Claims.

Adding fields to pages in the DCO affects the accuracy of Autofield unless steps are taken so that Autofield knows how to handle them.

Adding a field to a page

In addition, when adding fields to a page, the new fields will show up in the DCO in the exact order you arrange them in the hierarchy. However, these fields will be added to the end of the `setup.xml` file in the order that they are created. After adding any new fields to a page, you need to manually edit the `setup.xml` to put the fields in the correct order.

Adding a field that is not under a page

When you add a field to the DCO that you do not want to be recognized as a field on any page, you need to add the *Autofield* variable to the new field and set its value to -1.

Parent topic: [Data recognition](#)

TravelDocs: Specification of recognition zones

You must define the positions of the various fields for one variant of each page type. As Datacap Studio locates field zones that are on the fingerprint images, it writes the position information for each field into the document hierarchy.

- [Creating the text zones on the Rental_Agreement page](#)
Use Datacap Studio to create the text zones for the TravelDocs Rental_Agreement page.
- [Creating the OMR zones on the Rental_Agreement page](#)
You must create OMR zones to recognize check boxes on the page.
- [Creating the zones for the other page types](#)
Use Datacap Studio to create the zones for the other page types in the TravelDocs application.

Parent topic: [Data recognition](#)

Creating the text zones on the Rental_Agreement page

Use Datacap Studio to create the text zones for the TravelDocs Rental_Agreement page.

Procedure

To create the text zones on the Rental_Agreement page:

1. In the Fingerprints pane on the Datacap Studio Zones tab, select the first Rental_Agreement page (Car Rental #1).
2. In the Image View pane, click Zoom to enlarge the page so you can see the fields clearly.
3. In the Document hierarchy pane, click Lock DCO for editing.
4. In the Document hierarchy pane, expand the Rental_Agreement page and select the Pickup_Date field. Use the mouse to draw a bounding box around the pickup date on the page image.
5. Repeat for the Pickup_Location, Return_Date, Return_Location, Car_Type, and Total_Cost fields. Provide enough horizontal space around the field in case the text on the runtime page image is longer than the text on the fingerprint image.
6. In the Document hierarchy pane, click Save and then click Unlock DCO.
7. In the Document hierarchy pane, select any one of the fields you defined (for example, the Car_Type field). Then, look in the Properties pane at the lower left of the Zones tab. Make sure that the zone coordinates are saved in the field's Pos<id> variable, where <id> is the fingerprint you selected.

Parent topic: [TravelDocs: Specification of recognition zones](#)

Creating the OMR zones on the Rental_Agreement page

You must create OMR zones to recognize check boxes on the page.

About this task

The Rental_Agreement page type includes an Options field with three subfields:

- Nav_System
- Child_Seat
- Fuel_Service

These options are check box options that you manage by using optical mark recognition (OMR). OMR zones must always be subfields of a parent field. There are two approaches you can use:

- You can create a single parent field that contains all of the OMR subfields. This tutorial uses this technique for the rental agreement page.
- You can create a separate parent field for each OMR subfield. This tutorial will use this technique when you do the optional insurance page.

Procedure

To create the OMR zones on the rental agreement page:

1. In the Datacap Studio Zones tab Document Hierarchy pane, click Lock DCO.
2. In the Document Hierarchy pane, expand the Rental_Agreement page if necessary. Then, select the Options field and draw a bounding box around the Options region on the page image.
3. Expand the Options field node, select the Nav_System subfield, and draw a bounding box around the GPS Navigation check box. Then, repeat for the Child_Seat and Fuel_Service options. Try to make the bounding boxes as close to the same size as possible.
4. In the Document Hierarchy pane, click Save.

Parent topic: [TravelDocs: Specification of recognition zones](#)

Creating the zones for the other page types

Use Datacap Studio to create the zones for the other page types in the TravelDocs application.

About this task

Procedure

To create the zones for the other page type:

1. In the Fingerprints pane, select the first Optional_Insurance page (Car Rental #1).
2. In the Document hierarchy pane, expand the Optional_Insurance page. Select the CDW field, and draw a bounding box around the Collision Damage Waiver option on the page image.
3. Repeat for the PAI, PEP, and ELP fields. The size of the bounding boxes around the parent fields is not critical because it determines only what gets displayed to the operator during verification.
Attention: The parent field defines the region that is displayed during verification, so the dimensions are not critical.
4. In the Document hierarchy pane, expand the CDW field node. Select the CDW_Option subfield, and draw a bounding box around the Collision Damage Waiver check box on the page image. Try to make the bounding box the same size as the ones you drew on the rental agreement page.
5. Repeat for the PAI_Option, PEP_Option, and ELP_Option subfields. Try to make the bounding boxes around the check boxes the same size as the ones you drew for the options on the previous page.
6. In the Document hierarchy pane, click Save.
7. In the Fingerprints pane, select the first Air_Ticket page (Airline #1).
8. In the Document hierarchy pane, expand the Air_Ticket page. Then, select each of the fields in turn and draw a bounding box around the corresponding region on the page image. Then, click Save.
9. Repeat for each of the fields on the first Room_Receipt page (Hotel #1).

10. In the Document hierarchy pane, click Save and then click Unlock DCO.

Parent topic: [TravelDocs: Specification of recognition zones](#)

TravelDocs: Assignment of default rules to the document hierarchy

When the Datacap Studio application wizard generates the application framework, it attaches only default rules to default elements. Therefore, you must attach rules to the new elements that you created.

- [Assigning the default page level rules to new pages](#)

When you built the document hierarchy, you renamed the default page type from `Page` to `Rental_Agreement`, which therefore includes the default rules. However, you must assign rules for the other page types that you created.

- [Assigning the default field level rules to new fields](#)

When you built the document hierarchy, you renamed the default field type from `Field` to `Pickup_Date`, which therefore includes the default rules. However, you must assign rules for the other fields that you created.

- [Updating the Recognize Page rule](#)

The purpose of the Recognize Page rule is to locate the fields on each page and capture the data.

- [Running a batch through the workflow](#)

After you attach the default rules to the document hierarchy, you can run a batch through the workflow to view the progress of the application.

Parent topic: [Data recognition](#)

Assigning the default page level rules to new pages

When you built the document hierarchy, you renamed the default page type from `Page` to `Rental_Agreement`, which therefore includes the default rules. However, you must assign rules for the other page types that you created.

Procedure

To assign the default page level rules to the new pages:

1. On the Datacap Studio Rulemanager tab, in the Document Hierarchy pane, click Lock DCO.
2. Expand the document hierarchy so that you can see all the page nodes (`Rental_Agreement`, `Optional_Insurance`, `Air_Ticket`, and `Room_Receipt`).
3. In the Rulesets pane, expand each of the rulesets so you can see the rules that they contain (`VScan`, `ImageFix Load Settings`, `Enhance Image`, `PageID`, and so on).
4. In the CreateDocs ruleset, select the Create Fields rule.
5. In the Document Hierarchy pane, select the page node `Optional_Insurance`. Then, click Add to DCO on the side of the Rulesets pane. The Create Fields rule is added to the `Optional_Insurance` page's Open element.
6. Repeat for the `Air_Ticket` and `Room_Receipt` pages.
7. In the Recognize ruleset, select the Recognize Page rule. Then, add the rule to the `Optional_Insurance`, `Air_Ticket`, and `Room_Receipt` pages.
8. Repeat to add the Validate: Validate Page, Routing: Routing Rule 1, and Export: Export Page Fields rules to the same pages. Each of the pages now has an Open element.
9. Click Save.

Parent topic: [TravelDocs: Assignment of default rules to the document hierarchy](#)

Assigning the default field level rules to new fields

When you built the document hierarchy, you renamed the default field type from `Field` to `Pickup_Date`, which therefore includes the default rules. However, you must assign rules for the other fields that you created.

Procedure

To assign the default field level rule to the new fields:

1. In the Document Hierarchy pane, make sure that the DCO is still locked for editing.
2. Expand the document hierarchy so you can see all the field nodes (`Pickup_Date`, `Pickup_Location`, and others.).
3. In the Clean ruleset, select the Fields Clean rule.
4. In the Document Hierarchy pane, select the field node `Pickup_Location`. Then, click `Add to DCO` on the side of the Rulesets pane. The Fields Clean rule is added to the `Pickup_Location` field's `Open` element.
5. Repeat for the remaining fields. But do not add the rule to the `Options` field and its subfields on the rental agreement page, or to the fields on the optional insurance page. They are the container groups for the check box options. Also, the `Total_Cost` and `Return_Date` field definitions are shared across multiple pages. So you are only able to add the rule to the first instance of each field.
6. When you are finished, click `Save` and then click `Unlock DCO`.

Parent topic: [TravelDocs: Assignment of default rules to the document hierarchy](#)

Updating the Recognize Page rule

The purpose of the Recognize Page rule is to locate the fields on each page and capture the data.

About this task

The default Recognize Page rule uses the information in the document hierarchy to locate the field zones. It then completes text recognition on those zones by using `OCR_s` to extract the data.

In the TravelDocs application, you completed full page OCR during fingerprint creation and page identification. So, it is not necessary to do OCR again on the individual fields. Instead, you can use the `SnapCCOtoDCO` action to take the recognition data from the runtime fingerprint CCO file and apply it to the runtime batch hierarchy.

Procedure

To update the Recognize Page rule:

1. On the Datacap Studio Rulemanager tab, select the Recognize ruleset and click `Lock/Unlock ruleset` to lock the ruleset for editing.
2. Expand the Recognize ruleset completely.
3. Right-click the `RecognizePageFieldsOCR_S` action and choose `Remove`.
4. Click the Actions library tab.
5. Expand the `Recog_Shared` library and select `SnapCCOtoDCO`.
6. Make sure `Recognize: Page Function 1` is selected in the Rulesets pane.
7. Click `Add to function` at the left side of the Actions Library pane.
8. In the Rulesets pane, click `Save`. Then, click `Lock/Unlock ruleset` and choose `Publish Ruleset`.

Parent topic: [TravelDocs: Assignment of default rules to the document hierarchy](#)

Running a batch through the workflow

After you attach the default rules to the document hierarchy, you can run a batch through the workflow to view the progress of the application.

Procedure

1. Click the Datacap Studio Test tab.
2. In the Workflow pane, select the VScan task profile under Main Job.
3. Click New to start a new batch.
4. Click Process rules for target object on the main Test tab toolbar.
5. When you are prompted to release the batch, click Advance. The batch is moved to the next step in the workflow, which is PageID.
6. Click Process rules for target object on the main Test tab toolbar and wait while the task profile launches. It might take a few moments as Datacap must run full page OCR on all the images in the images folder.
7. When asked if you want to release the batch, click Advance. The batch is moved to the next step in the workflow, which is Profiler.
8. Click Process rules for target object on the main Test tab toolbar and wait while the Profiler task profile runs.
9. When asked if you want to release the batch, click Advance. The batch is moved to the next step in the workflow, which is Verify.
10. On the Runtime batch hierarchy tab, expand the first car rental agreement page to see each of the defined fields and the associated data.
Attention: The Pickup_Date field in one of the sample images includes an intentional blurred character so that you can examine recognition confidence levels.
11. Because you have more work to do before you are ready to run the Verify task profile, right-click the batch in the Workflow pane and choose Cancel.

Parent topic: [TravelDocs: Assignment of default rules to the document hierarchy](#)

TravelDocs: Updating the application to manage check box options

You need to configure required variables, specify the check mark type, and create a rule to recognize the OMR fields to manage check box options.

- [Setting the required variables on the Options and Insurance fields](#)
When you are updating the TravelDocs application, you must use Datacap Studio to set the required variables on the Options and Insurance fields.
- [Specifying the check mark type](#)
When you use the OCR_A engine for check mark recognition, you must specify whether the check marks have bounding boxes.
- [Creating a rule to recognize the OMR fields](#)
You must use Datacap Studio to create the rule to recognize OMR fields.
- [Adding the Recognize OMR Fields rule to the document hierarchy](#)
You must add the Recognize OMR fields rule to the document hierarchy so that Datacap can recognize check marks.
- [Running a batch through the workflow](#)
You can run a batch through the workflow in Datacap Studio to determine if the application recognizes check marks.

Setting the required variables on the Options and Insurance fields

When you are updating the TravelDocs application, you must use Datacap Studio to set the required variables on the Options and Insurance fields.

Procedure

To set variables on the Options and Insurance fields:

1. On the Datacap Studio Rulemanager tab in the Document Hierarchy pane, click Lock DCO.
2. Expand the Car_Rental document and the Rental_Agreement page.
3. Right-click the Options field and choose Manage variables.
4. Click New, type RecogType, and press Enter. Then, click New, type MultiPunch, and press Enter.
Attention: Variables are case-sensitive, so make sure that you capitalize RecogType and MultiPunch as shown here.
5. Enter the values RecogType=4 and MultiPunch=1. Then, click Done.
6. Expand the Optional_Insurance page.
7. Right-click the CDW field and choose Manage variables.
8. Click New, type RecogType, and press Enter. MultiPunch is not required because the parent field has only one subfield.
9. Enter the value RecogType=4 and click Done.
10. Repeat for the other parent fields (PAI, PEP, and ELP).
11. In the Document Hierarchy pane, click Save and then click Unlock DCO.

Parent topic: [TravelDocs: Updating the application to manage check box options](#)

Specifying the check mark type

When you use the OCR_A engine for check mark recognition, you must specify whether the check marks have bounding boxes.

About this task

The default setting for check marks is for no bounding box (Clear Background). Because all of the check marks in the sample images have square bounding boxes, you must change the Checkmark Type setting in on the OCR/A settings tab. The settings tabs for the various recognition engines are displayed along the bottom of the Properties pane on the Datacap Studio Zones tab.

Procedure

1. Click the Datacap Studio Zones tab.
2. In the Document Hierarchy pane, click Lock DCO.
3. Expand the Rental_Agreement page and select the Options field.
4. Look at the bottom of the Properties pane. If the OCR/A tab is not visible, right-click any existing tab and choose Show tabs and select the OCR/A option.
5. Click the OCR/A tab.
6. In the Properties pane, scroll down to the OMR section and set the Checkmark type to Square background.

7. In the Document Hierarchy pane, click Save.
8. Expand the Optional_Insurance page and do the same for the CDW, PAI, PEP, and ELP fields.
9. In the Document Hierarchy pane, click Save and then click Unlock DCO.

Parent topic: [TravelDocs: Updating the application to manage check box options](#)

Creating a rule to recognize the OMR fields

You must use Datacap Studio to create the rule to recognize OMR fields.

About this task

Procedure

1. In the Rulesets pane, select the Recognize ruleset and click Lock/Unlock ruleset (padlock) to lock the ruleset for editing.
2. Right-click the Recognize ruleset and choose Add Rule.
3. Rename the new rule from Rule1 to Recognize OMR Fields.
4. Rename the default function from Function1 to Recognition: OMR.
5. Click the Actions library tab.
6. Expand the ocr_a library and select RecognizeFieldOCR_A.
7. Make sure the Recognition:OMR function is selected in the Rulesets pane.
8. Click Add to function at the left side of the Actions Library pane.
9. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and choose Publish Ruleset.

Parent topic: [TravelDocs: Updating the application to manage check box options](#)

Adding the Recognize OMR Fields rule to the document hierarchy

You must add the Recognize OMR fields rule to the document hierarchy so that Datacap can recognize check marks.

Procedure

To add the Recognize OMR Fields rule to the document hierarchy:

1. In the Document Hierarchy pane, click Lock DCO.
2. Expand the document hierarchy so the Options field on the Rental_Agreement page and CDW, PAI, PEP, and ELP parent fields on the Optional_Insurance page are visible.
3. In the Recognize rule set, select the Recognize OMR Fields rule.
4. In the Document Hierarchy pane, select the Options field node. Then, click Add to DCO on the Rulesets pane. The Recognize OMR Fields rule is added to the Open element of the Options field.
5. Repeat to add the rule to each of the parent fields on the Optional_Insurance page.
6. In the Document Hierarchy pane, click Save and then click Unlock DCO.

Parent topic: [TravelDocs: Updating the application to manage check box options](#)

Running a batch through the workflow

You can run a batch through the workflow in Datacap Studio to determine if the application recognizes check marks.

Procedure

1. Run a batch through the VScan, Page ID, and Profiler task profiles as described earlier (see [Running a batch through the workflow](#)).
2. When the Profiler task completes and you advance the batch to the Verify task, expand the first car rental page on the Runtime batch hierarchy tab. You can see the fields and the associated data. The value of the Options field is 001, which indicates that Datacap interpreted the first option as `Not selected`, the second option as `Not selected`, and the third option as `Selected`.
3. Expand the first optional insurance page and confirm that the values are correct.
4. Right-click the batch in the Workflow pane and choose Cancel.

Parent topic: [TravelDocs: Updating the application to manage check box options](#)

TravelDocs: Using pixel threshold check box recognition (optional)

For demonstration purposes, you can go through the pixel threshold method.

- [Updating the Recognize OMR Fields rule to use RecognOMRThreshold](#)
You must use Datacap Studio to update any rule, including the Recognize OMR Fields rule.
- [Determining appropriate threshold and background settings](#)
The parameters on the RecognOMRThreshold action include a pair of numerical values. The first number is the threshold parameter and the second number is the background parameter. Obtaining optimum values requires that you run multiple sample pages through the workflow and then check the density strings and confidence levels.

Parent topic: [Data recognition](#)

Updating the Recognize OMR Fields rule to use RecognOMRThreshold

You must use Datacap Studio to update any rule, including the Recognize OMR Fields rule.

About this task

Procedure

To update the Recognize OMR Fields rule to use RecognOMRThreshold:

1. In the Rulesets pane, select the Recognize ruleset and click Lock/Unlock ruleset to lock the ruleset for editing.
2. Expand the Recognize ruleset, the Recognize OMR Fields rule, and the Recognition: OMR function.
3. Right-click the RecognizeFieldOCR_A action and choose Remove.
4. Click the Actions library tab.
5. Expand the Recog_Shared library and select RecognOMRThreshold.
6. Make sure the Recognition:OMR function is selected in the Rulesets pane.
7. Click Add to function at the left side of the Actions Library pane.
8. Select the RecognOMRThreshold action and set the parameters in the Properties pane as follows:

- Threshold = 20
- Background = 20

Attention: These values are the starting values, which you can later recalculate with more accurate values.

9. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and choose Publish Ruleset.

Parent topic: [TravelDocs: Using pixel threshold check box recognition \(optional\)](#)

Determining appropriate threshold and background settings

The parameters on the `RecogOMRThreshold` action include a pair of numerical values. The first number is the threshold parameter and the second number is the background parameter. Obtaining optimum values requires that you run multiple sample pages through the workflow and then check the density strings and confidence levels.

- [Checking the option values and obtaining the density string values](#)
To check the option values and obtain the density string values, you must run a batch in Datacap Studio. Then, open the page data file in the runtime batch folder.
- [Interpreting the density string values](#)
By using the density string characters and the corresponding formula, you can calculate the percentage of black pixels for each OMR zone.

Parent topic: [TravelDocs: Using pixel threshold check box recognition \(optional\)](#)

Checking the option values and obtaining the density string values

To check the option values and obtain the density string values, you must run a batch in Datacap Studio. Then, open the page data file in the runtime batch folder.

Procedure

1. Run a batch through the VScan, Page ID, and Profiler task profiles as described in the [Running a batch through the workflow](#) topic.
2. When the Profiler task is completed and you advance the batch to the Verify task, expand the first car rental page on the Runtime batch hierarchy tab. You see the fields and the associated data. The value of the `Options` field is `001`, indicating that Datacap interpreted the first option as `not selected`. Datacap interpreted the second option as `not selected`, and the third option as `selected`.
3. Select the `Options` field in the runtime batch hierarchy. Then, click the `Image` tab in the center pane of Datacap Studio if it is not already visible. The `Options` zone is highlighted. The first and second options are not selected, and the third is selected. In this case Datacap determined the values correctly. However, depending on the amount of space around the check boxes, your version might not work. Even if the 20,20 values worked, you must continue to determine optimum parameter values.
4. Open the most recent runtime batch folder in `C:\Datacap\TravelDocs\batches`.
5. Open the file `tm000001.xml` and scroll almost to the bottom to see the `Options` field data.
6. Make a note of the `DensityString` value (`ACG` in this example) and then close the file.
Important: Note the three lines that follow the density string line. These lines represent the three check box options. The values `48` and `49` are the ASCII values for `0` (not selected) and `1` (selected). The `cn` attributes represent the confidence level, where `10` is high confidence and `5` is low confidence.
7. Repeat the steps to check the values for the Insurance options and make a note of the `DensityString` value in `tm000002.xml`.

From tm000002.xml:

```
CDW: <V n="DensityString">C</V>
PAI: <V n="DensityString">C</V>
PEP: <V n="DensityString">B</V>
ELP: <V n="DensityString">G</V>
```

Parent topic: [Determining appropriate threshold and background settings](#)

Interpreting the density string values

By using the density string characters and the corresponding formula, you can calculate the percentage of black pixels for each OMR zone.

The previous examples produce these results.

Parent field	Subfield	Checkbox	Density value	Percentage filled
Options	Nav_System	Cleared	A	17%
	Child_Seat	Cleared	C	19%
	Fuel_Service	Checked	G	23%
Insurance	CDW	Cleared	C	19%
	PAI	Cleared	C	19%
	PEP	Cleared	B	18%
	ELP	Checked	G	23%

By using the data in first table, you can determine that $\text{threshold} = 20.5$ and $\text{background} = 20$ represent good values. The $(2 * \text{threshold} - \text{background})$ formula sets the high-confidence threshold at 21, but you can scan more pages and check their density strings before you set final values.

Parent field	Subfield	Checkbox	Percentage filled	Result in runtime hierarchy
Options	Nav_System	Cleared	17%	Option: Selected Confidence: High
	Child_Seat	Cleared	19%	Option: Not selected Confidence: High
	Fuel_Service	Checked	23%	Option: Selected Confidence: High
Insurance	CDW	Cleared	19%	Option: Selected Confidence: High
	PAI	Cleared	19%	Option: Not selected Confidence: High
	PEP	Cleared	18%	Option: Not selected Confidence: High
	ELP	Checked	23%	Option: Not selected Confidence: High

Because the OMR zones that you drew are probably different from the ones that generated the data in second table, your values are different. Use the density characters that you obtained to determine appropriate values

for the threshold and background parameters, and update the Recognize OMR Fields rule. Then, run a new batch and review the results, including the confidence values in the page data files.

Parent topic: [Determining appropriate threshold and background settings](#)

Data Validation

Data validation determines whether captured data complies with the data integrity rules that are defined in your business requirements.

For example, when you established the business rules for the TravelDocs application, you tested if the cost fields are in a valid currency format. If the car type on the rental agreement page is one of a set of predefined values. And if the total cost on the air ticket page equals the air fare plus taxes.

A validation failure does not necessarily mean that the original page contains invalid data. It might mean that the recognition engine failed to recognize one or more characters correctly. You can change the status of the page, which contains the data that failed validation. This change ensures that the page is displayed to an operator for verification.

- [Validate the data](#)
Datacap completes validation by using rules that you create and attach to specific items in the document hierarchy.
- [TravelDocs: Update the application to complete validation](#)
You can update your application to complete validation by using actions, external data sources, or dictionaries.

Parent topic: [Datacap application development](#)

Validate the data

Datacap completes validation by using rules that you create and attach to specific items in the document hierarchy.

The purpose of validation is to determine whether captured data conforms to specified business rules. For example:

- Does an expense lie within permitted limits?
- Are dates valid and within a permitted range?
- Is the total cost calculated correctly?
- Does the vendor information match the information that is stored in a database of approved vendors?
- Does a field value match one of a set of permitted values?

To check whether an expense lies within permitted limits, you might first create a rule that does the following actions.

- Ensures that the expense field contains numbers in a valid currency format
- Determines whether the value is less than or equal to the maximum permitted limit
- Does exception handling if the value is invalid or above the permitted limit

You can then attach the rule to the expense field in the document hierarchy.

- [Check data format validity](#)
Before, you can apply specific business rules to a field, you must confirm that the data format is valid.
- [Validate calculated fields](#)
You can run validation to ensure that a calculated value is correct.

- [Show validation failures to an operator](#)
To determine the pages that are to be displayed to an operator, Datacap maintains a *Status* variable for each object in the runtime batch hierarchy. For example, a status of 0 indicates that operations on the object were successful while a status code of 1 indicates a problem or potential problem.
- [Use external data sources during validation](#)
You might need to compare runtime field values to an external data source to determine whether the values on a page are valid. For example, you might need to determine whether the vendor information matches the information that is stored in a database of approved vendors.
- [Manage validation errors](#)
Validation actions set the *Status* variable of the object if there is an error. The *Status* variable determines which pages Datacap displays to the operator. However, you might need to complete extra error handling within the application.

Parent topic: [Data Validation](#)

Check data format validity

Before, you can apply specific business rules to a field, you must confirm that the data format is valid.

For example, you cannot test whether an expense lies within allowed limits until you determine that the field contains a valid currency value.

The business requirements of the application specify valid formats for all of the fields that your application is testing. The following examples describe the values of three acceptable currencies as defined by the business requirements for the TravelDocs application.

- \$477.82
- 824.83
- 254.40 USD

The Validations actions library includes several actions that test the format of a field, including.

Library	Action	Description
Validations	IsFieldCurrency	Returns True if the field is a number and includes a two-digit decimal amount; returns False otherwise. The action ignores any leading currency symbol (for example, \$).
Validations	IsFieldDate	Returns True if the field is in one of the supported date formats; returns False otherwise.
Validations	IsFieldLengthMax	Returns True if the field contains no more than the specified number of characters; returns False otherwise.
Validations	IsFieldLengthMin	Returns True if the field contains at least the specified number of characters; returns False otherwise.
Validations	IsFieldPercentAlpha	Returns True if the field contains no numbers or special characters; returns False otherwise.

Library	Action	Description
Validations	IsFieldPercentNumeric	Returns True if the field contains no letters or special characters; returns False otherwise.

For detailed information on these and other actions in the Validations library, select the action on the Actions Library tab and click Display information.

Parent topic: [Validate the data](#)

Validate calculated fields

You can run validation to ensure that a calculated value is correct.

A calculated field obtains its value from one or more independent fields. For example, on the TravelDocs air ticket, the total cost equals the airfare plus taxes and fees.

In the air ticket example, you can use validation to ensure that the total cost is correct, based on the airfare and tax fields. An error does not necessarily mean that the value was calculated incorrectly on the original page. It might mean that one of the values was recognized incorrectly. You can change the status of the page, which contains the data that failed validation. This change ensures that the page is displayed to an operator for verification.

The Validations actions library includes a CalculateFields action completes arithmetic operations on number field values.

Library	Action	Description
Validations	CalculateFields	Returns True if the arithmetic expression is valid; returns False otherwise.

The following example returns `True` if the total cost equals the airfare plus taxes.

```
CalculateFields("'Total_Cost' = 'Airfare' + 'Taxes'")
```

The fields that you reference must contain number data. Otherwise, the action fails.

Because a calculation involves multiple field values, and because the `CalculateFields` action operates only on child objects, you cannot complete the calculation at the field level. Instead, you must complete the calculation at the page level.

Important: You can complete calculations on values from different pages or different documents within the batch. To do so, you must use variables.

When to do validation on calculated fields

The `CalculateFields` action works only on number fields. If your Validation rule set includes rules to validate and possibly correct the format of individual fields (for example, removing a "USD" suffix from a currency field). You must run the page-level validation after you completed all field-level validations. For example, an airline ticket has an associated series of fields. You can validate this associated series of fields in sequence before a full page-level validation by using the `Close` element of the page.

- Air_Ticket
- Open
 - Item Cost

- Open
 - (global)
 - Validate: Validate Currency field
 - Close
- Taxes
 - Open
 - (global)
 - Validate: Validate Currency field
 - Close
- Total Cost
 - Open
 - (global)
 - Validate: Validate Currency field
 - Close
- Close
- (global)
 - Validate: Validate Total Cost

The `Validate Total Cost` rule runs after Datacap finishes processing all of the child fields on the page.

Parent topic: [Validate the data](#)

Show validation failures to an operator

To determine the pages that are to be displayed to an operator, Datacap maintains a *Status* variable for each object in the runtime batch hierarchy. For example, a status of 0 indicates that operations on the object were successful while a status code of 1 indicates a problem or potential problem.

Datacap updates the status when rules are started.

Datacap uses other status codes, such as 49, which indicates that a page was scanned successfully. By default, Datacap displays all pages, but you can configure the TravelDocs application to display only pages with a status of 1. For more information, see the topic [Page status](#).

The following problems result in a status code of 1.

- Unrecognized or low confidence characters: By default, if a page has any low confidence characters, Datacap sets the page status to 1. Fields with low confidence characters are displayed in yellow in the verify panel.
- Validation failures: If a field fails validation, Datacap sets the field status to 1 and the page status to 1 and displays the field in red in the verify panel.
 Important: When there is a validation failure, Datacap sets all parent objects' status to 1, including the batch object. For example, if a subfield fails, Datacap sets the parent field to 1. The parent of the field (the page) is set to 1. The parent document of the page is set to 1. The parent batch object status is set to 1. There is only one batch object in a batch.

Use of the `Status_Preserve_OFF` action in the rrunner library to set the status correctly. This action and the related `Status_Preserve_ON` action determine whether validation rules can update an object's status.

Library	Action	Description

Library	Action	Description
rrunner	Status_Preserve_OFF	Turns the Status Preserve setting of a page and its child fields to OFF, meaning validation rules can update an object's status if a validation fails.
rrunner	Status_Preserve_ON	Turns the Status Preserve setting of a page and its child fields to ON, meaning validation rules cannot change an object's status.

In most situations, you want to make sure Status Preserve is turned OFF at the start of validation.

The default page-level rule Validate Page in the Validate ruleset that is generated by the Application wizard sets Status Preserve to OFF for you.

The rule is attached to the default page type, but you must attach it manually to any new pages that you create.

Reading the status variable

You cannot check the status variable for a page or field from within Datacap Studio. So, you must read the runtime batch data files:

- You can get the status of each page from the task profile's data file (for example, Profiler.xml).
- You can get the status of each field from the page data files (for example, tm000001.xml).

Profiler.xml (page status)	tm000001.xml (field status)
<pre><P id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev114_TM000001"> <V n="TYPE">Rental_Agreement</V> <V n="STATUS">1</V> <V n="IMAGEFILE">tm000001.tif</V> etc. </P></pre>	<pre><F id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev114_Pickup_Date"> <V n="TYPE">Pickup_Date</V> <V n="Position">194,402,563,458</V> <V n="STATUS">0</V> <C cn="10" cr="203,416,225,438">77</C> <C cn="10" cr="230,423,245,438">111</C> etc. </F></pre>

In Profiler.xml, `<P id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev114_TM000001">` is the page definition. In tm000001.xml, `<F id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev114_Pickup_Date">` is the field definition.

Parent topic: [Validate the data](#)

Use external data sources during validation

You might need to compare runtime field values to an external data source to determine whether the values on a page are valid. For example, you might need to determine whether the vendor information matches the information that is stored in a database of approved vendors.

The Lookup library includes actions for connecting to external data sources and running SQL statements. The available actions include the following.

Library	Action	Description
Lookup	OpenConnection	Uses a data source name or connection string to open a connection to a database.
Lookup	ExecuteSQL	Runs a SQL statement. Returns True if the SQL statement runs successfully and any SELECT statement returns a value.
Lookup	CloseConnection	Closes an open database connection.

For an example of how to use an external data source to complete validation, see the [Use a lookup database to validate the car type](#) topic.

Parent topic: [Validate the data](#)

Manage validation errors

Validation actions set the `Status` variable of the object if there is an error. The `Status` variable determines which pages Datacap displays to the operator. However, you might need to complete extra error handling within the application.

The following rule has two functions and each function has two actions. Suppose that `Function1` contains validation actions and `Function2` is there to handle validation errors.

```
Rule: Validation rule
  Function1: Perform validation
    Action1
    Action2
  Function2: Handle validation errors
    Action3
    Action4
```

Using the rules for execution of functions and actions within a rule:

- If `Action1` returns `False`, Datacap skips `Action2` and runs `Function2`.
- If `Action1` returns `True` and `Action2` returns `False`, Datacap runs `Function2`.
- If `Action1` and `Action2` both return `True`, Datacap does not run `Function2`.

Parent topic: [Validate the data](#)

TravelDocs: Update the application to complete validation

You can update your application to complete validation by using actions, external data sources, or dictionaries.

After you create a batch with the updated application, you can examine status values, and create more recognition zones.

- [Validate the currency fields](#)
To validate currency fields, you need to create a rule with actions that delete letters and spaces, and confirm that the field contains a currency value. Then, the rule must be assigned to all of the currency fields in the document hierarchy.

- [Validate the flight cost](#)
To validate the flight cost, you need to create a rule that calculates the cost field. Then, you need to assign the rule to the Air_Ticket page `Close` element in the document hierarchy.
- [Use a lookup database to validate the car type](#)
The rental agreement page includes the car_type field, which you can test to determine whether the fields contains a permitted value.
- [Creating a dictionary of valid car types](#)
If there is a problem with the car_type field in the verification panel, Datacap can present a list of valid car types from which the operator can select a valid type.
- [Running a batch through the workflow](#)
After you create the dictionary and attach it to the Car_Type field, you can run a batch through the workflow to see how the application is progressing.
- [Examination of page and field status values](#)
The validation rules affect the status that Datacap assigns to the status variable for each page and field. To see the page status, open Profiler.xml in the application's most recent batch folder.
- [Creating recognition zones for the remaining fingerprints](#)
After you review the page status and field status and confirm that the application is working properly, create the recognition zones for the remaining fingerprints.
- [Running a batch through the workflow](#)
After you define all of the required recognition zones, you can run a batch through the workflow.
- [Page and field status codes in the TravelDocs application](#)
After you run a batch through the workflow, review the status codes for each of the fields that you validated and for the pages in the runtime batch.

Parent topic: [Data Validation](#)

Validate the currency fields

To validate currency fields, you need to create a rule with actions that delete letters and spaces, and confirm that the field contains a currency value. Then, the rule must be assigned to all of the currency fields in the document hierarchy.

The currency field validation rule that you need to create uses the following actions.

Action	Description
DeleteAllAlpha	Deletes all of the letters from the field. Use this action to remove the USD suffix that are used by one of the airlines.
DeleteSelectedChars	Deletes the specified characters from the field. Use this action to remove any spaces.
IsFieldCurrency	Returns True if the field is a number and includes a two-digit decimal amount; returns False otherwise. The action ignores any leading currency symbol (for example, \$).

- [Creating the Validate Currency Field rule](#)
You create the Validate Currency Field rule in Datacap Studio.
- [Adding the Validate Currency Field rule to the document hierarchy](#)
To add the Validate Currency Field rule to the document hierarchy, you must use Datacap Studio

Parent topic: [TravelDocs: Update the application to complete validation](#)

Creating the Validate Currency Field rule

You create the Validate Currency Field rule in Datacap Studio.

Procedure

To create the Validate Currency Field rule:

1. Open Datacap Studio and click the Rulemanager tab.
2. In the Rulesets pane, select the Validate ruleset and click Lock/Unlock ruleset (padlock) to lock the ruleset for editing.
3. Right-click the Validate ruleset and choose Add Rule.
4. Rename the new rule from Rule1 to Validate Currency Field.
5. Rename the default function from Function1 to Validation: Currency.
6. Click the Actions library tab.
7. Expand the Validations library and select the DeleteAllAlpha action.
8. Make sure the Validation: Currency function is selected in the Rulesets pane.
9. Click Add to function at the left side of the Actions library pane.
10. On the Actions library tab, select the DeleteSelectedChars action and click Add to Function.
11. On the Actions library tab, select the IsFieldCurrency action and click Add to Function.
12. Select the DeleteSelectedChars action and set the strParam parameter to ' ' (a single space) in the Properties pane.
13. In the Rulesets pane, click Save.

Parent topic: [Validate the currency fields](#)

Adding the Validate Currency Field rule to the document hierarchy

To add the Validate Currency Field rule to the document hierarchy, you must use Datacap Studio

Procedure

To add the Validate Currency Field rule to the document hierarchy:

1. In the Document Hierarchy pane, click Lock DCO.
2. Expand the Car_Rental > Rental_Agreement page so that the fields are visible.
3. In the Validate ruleset, select the Validate Currency Field rule.
4. In the Document Hierarchy pane, select the Total_Cost field node. Then, click Add to DCO on the left side of the Rulesets pane. The Validate Currency Field rule is added to the Open element in the Total Cost field.
5. Expand the Flight > Air_Ticket page so the fields are visible.
6. Select the Airfare field and click Add to DCO to add the Validate Currency Rule to the Open element of the Airfare field.
7. Select the Taxes field and click Add to DCO to add the Validate Currency Rule to the Open element of the Taxes field. Because the Total_Cost field definition is shared across pages, the Validate Currency Rule is already included in the Total_Cost field on the Air_Ticket and Room_Receipt pages.
8. In the Document Hierarchy pane, click Save.

Parent topic: [Validate the currency fields](#)

Validate the flight cost

To validate the flight cost, you need to create a rule that calculates the cost field. Then, you need to assign the rule to the Air_Ticket page `Close` element in the document hierarchy.

The rule that you create to validate the total cost field uses this action.

Action	Description
CalculateFields	Returns True if the arithmetic expression is valid; returns False otherwise.

- [Creating the Validate Flight Cost rule](#)
You create the Validate Flight Cost rule in Datacap Studio.
- [Adding the Flight Cost rule to the document hierarchy](#)
To add the Flight Cost rule to the document hierarchy, you must use Datacap Studio.

Parent topic: [TravelDocs: Update the application to complete validation](#)

Creating the Validate Flight Cost rule

You create the Validate Flight Cost rule in Datacap Studio.

Procedure

To create the Validate Flight Cost rule:

1. Make sure the Validate ruleset is locked for editing.
2. Right-click the Validate ruleset and choose Add Rule.
3. Rename the new rule from Rule1 to Validate Flight Cost.
4. Rename the default function from Function1 to Validation: Flight Cost.
5. On the Actions library tab, select the CalculateFields action (also in the Validations library).
6. Make sure the Validation: Flight Cost function is selected in the Rulesets pane.
7. Click Add to function at the side of the Actions Library pane.
8. Select the CalculateFields action and in the Properties pane set the strParam parameter to 'Airfare' + 'Taxes' = 'Total_Cost'. These parameters are the names of the fields as specified in the document hierarchy.
9. In the Rulesets pane, click Save.

Parent topic: [Validate the flight cost](#)

Adding the Flight Cost rule to the document hierarchy

To add the Flight Cost rule to the document hierarchy, you must use Datacap Studio.

Procedure

To add the Flight Cost rule to the document hierarchy

1. In the Document Hierarchy pane, make sure that the document hierarchy is locked for editing.
2. Select the Close element at the end of the Air_Ticket page definition.
3. In the Validate ruleset, select the Validate Flight Cost rule.
4. Click Add to DCO to add the Validate Flight Cost rule to the Close element of the Air_Ticket page.
5. In the Document Hierarchy pane, click Save.

Parent topic: [Validate the flight cost](#)

Use a lookup database to validate the car type

The rental agreement page includes the `car_type` field, which you can test to determine whether the fields contains a permitted value.

The list of permitted car types includes these types.

- Compact
- Standard
- Full size
- SUV
- Other

For each car type field, you test the field value and set the field status to 1. There is a problem if the car type does not match one of the permitted types.

To use the lookup database, use these actions.

Library	Action	Description
Lookup	OpenConnection	Uses a data source name or connection string to open a connection to a database.
Lookup	ExecuteSQL	Runs a SQL statement. Returns True if the SQL statement runs successfully and any SELECT statement returns a value.
Lookup	CloseConnection	Closes an open database connection.

- [Creating the lookup database table](#)
The Datacap installation includes a Microsoft Access lookup database, `TravelDocsLook.mdb`, which is in the `C:\Datacap\TravelDocs` folder. You create a table that is named `Car Types` in the lookup database.
- [Creating the Validate Car Type rule](#)
You create the Validate Car Type rule in Datacap Studio.
- [Adding the Validate Car Type rule to the document hierarchy](#)
To add the Validate Car Type rule to the document hierarchy, you must use Datacap Studio.

Parent topic: [TravelDocs: Update the application to complete validation](#)

Creating the lookup database table

The Datacap installation includes a Microsoft Access lookup database, `TravelDocsLook.mdb`, which is in the `C:\Datacap\TravelDocs` folder. You create a table that is named `Car Types` in the lookup database.

Procedure

To create the lookup database table:

1. Open the file `C:\Datacap\TravelDocs\TravelDocsLook.mdb` in Microsoft Access.
2. Create a table that is named `Car Types`.
3. Create a field that is named `Car_Type` of type `Text`.
4. Enter the permitted values: `Compact`, `Standard`, `Full size`, `SUV`, and `Other`.
5. Save the new table.

Parent topic: [Use a lookup database to validate the car type](#)

Creating the Validate Car Type rule

You create the Validate Car Type rule in Datacap Studio.

Procedure

To create the Validate Car Type rule:

1. In the Rulesets pane, make sure that the Validate ruleset is locked for editing.
2. Right-click the Validate ruleset and choose Add Rule.
3. Rename the new rule from Rule1 to Validate Car Type.
4. Rename the default function from Function1 to Validation: Car Type.
5. On the Actions library tab, expand the Lookup library and select the OpenConnection action.
6. Make sure the Validation: Car Type function is selected in the Rulesets pane.
7. Click Add to function at the left side of the Actions Library pane.
8. Select the ExecuteSQL action and click Add to Function.
9. Select the CloseConnection action and click Add to Function.
10. Select the OpenConnection action and in the Properties pane set the strParam parameter to `@APPVAR (* /lookupdb:cs)`.

This parameter is a Datacap smart parameter that obtains the connection string for the application's lookup database from the application configuration file.

11. Select the ExecuteSQL action and in the Properties pane set the sStringIn parameter to `"SELECT Car_Type FROM Car_Types WHERE Car_Type='%s';",Car_Type`.
Attention: You must use the syntax exactly as it is used here. You can copy and paste from here if necessary.
12. In the Rulesets pane, click Save. Then, click Lock ruleset and choose Publish ruleset.

Parent topic: [Use a lookup database to validate the car type](#)

Adding the Validate Car Type rule to the document hierarchy

To add the Validate Car Type rule to the document hierarchy, you must use Datacap Studio.

Procedure

To add the Validate Car Type rule to the document hierarchy:

1. In the Document Hierarchy pane, make sure that the document hierarchy is locked for editing.
2. Expand the Car_Rental > Rental_Agreement page so the fields are visible.
3. In the Validate ruleset, select the Validate Car Type rule.
4. In the Document Hierarchy pane, select the Car_Type field node. Then, click Add to DCO on the left side of the Rulesets pane. The Validate Car Type rule is added to the Open element of the Car_Type field.
5. In the Document Hierarchy pane, click Save.

Parent topic: [Use a lookup database to validate the car type](#)

Creating a dictionary of valid car types

If there is a problem with the `car_type` field in the verification panel, Datacap can present a list of valid car types from which the operator can select a valid type.

About this task

The Datacap Desktop and Datacap Web Client verification interfaces enable the population of a drop-down list directly from the database by using an SQL statement that is embedded in the `SELECT` variable of the field. You need to create the variable in Datacap Studio by first unlocking the document hierarchy, right-clicking the `Car_Type` field, and choosing Manage Variables. You can then add the `SELECT` variable and set it to the following value.

```
<SQL flist='Car_Type' dsn="*/lookupdb:cs">SELECT Car_Type FROM Car_Types</SQL>
```

An SQL query (`SELECT <column> FROM <table>`) gets the list of valid car types from the application's lookup database (`dsn="*/lookupdb:cs"`). It then creates a drop-down list in the specified field (`flist='<field>'`) that contains the returned values.

Another variable, *Lookup*, is functionally similar, except that it displays the list of available choices in a window instead of a drop-down list.

To create a selection list that works for all verification interfaces, you can create a dictionary that contains the same valid car types (Compact, Standard, Full size, SUV, and Other). You can then attach the library to the `Car_Type` field.

- [Creating the dictionary](#)
You create the dictionary in Datacap Studio.
- [Attaching the dictionary to the Car_Type field](#)
After you create the dictionary, you need to attach it to the `Car_Type` field in Datacap Studio.

Parent topic: [TravelDocs: Update the application to complete validation](#)

Creating the dictionary

You create the dictionary in Datacap Studio.

About this task

Procedure

To create the dictionary:

1. Confirm that the document hierarchy is locked for editing.
2. Click Dictionaries at the top of the Document Hierarchy pane.
3. Click Edit dictionary and choose Add dictionary.
4. Change the dictionary name from `<new_dictionary>` to `Car_Types`.
5. Right-click the new dictionary and choose `Add word`.
6. Change the name from `<new word>` to `Compact` and the value from `value` to `Compact`.
7. Repeat to add `Standard`, `Full size`, `SUV`, and `Other` to the dictionary.
8. Click Save.

Parent topic: [Creating a dictionary of valid car types](#)

Attaching the dictionary to the Car_Type field

After you create the dictionary, you need to attach it to the Car_Type field in Datacap Studio.

About this task

Procedure

To attach the dictionary to the Car_Type field:

1. Make sure that the document hierarchy is locked for editing.
2. Expand the Car_Rental > Rental_Agreement page so the fields are visible.
3. Right-click the Car_Type field and select Manage variables.
4. Click New, type `DICT`, and press Enter.
Important: Variables are case-sensitive. Ensure that you capitalize `DICT`.
5. Enter the value `Car_Types`. Then, click Done.
6. In the Document Hierarchy pane, click Save and then click Unlock DCO.

Parent topic: [Creating a dictionary of valid car types](#)

Running a batch through the workflow

After you create the dictionary and attach it to the Car_Type field, you can run a batch through the workflow to see how the application is progressing.

Procedure

1. Click the Datacap Studio Test tab.
2. In the Workflow pane, select the VScan task profile under Main Job.
3. Click New to start a new batch.
4. Click Process rules for target object on the main Test tab toolbar. When you are prompted to release the batch, click Advance. The batch is moved to the next step in the workflow, which is PageID.
5. Click Process rules for target object on the main Test tab toolbar. When you are prompted to release the batch, click Advance. The batch is moved to the next step in the workflow, which is Profiler.
6. Click Process rules for target object on the main Test tab toolbar and wait while the task profile launches. When you are prompted to release the batch, click Advance. The batch is moved to the next step in the workflow, which is Verify.
7. Because you are not ready yet to run the Verify task profile, right-click the batch in the Workflow pane and choose Cancel.

Parent topic: [TravelDocs: Update the application to complete validation](#)

Examination of page and field status values

The validation rules affect the status that Datacap assigns to the status variable for each page and field. To see the page status, open Profiler.xml in the application's most recent batch folder.

The Profiler.xml file includes the status of each page in the batch.

```
<D
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_2010
0323.007.01">
  <V n="TYPE">Car_Rental</V>
  <V n="STATUS">0</V>
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_TM00
```

```

0001">
  <V n="TYPE">Rental_Agreement</V>
  <V n="STATUS">1</V>
  etc.
</P>
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_TM00
0002">
  <V n="TYPE">Optional_Insurance</V>
  <V n="STATUS">0</V>
  etc.
</P>
</D>
<D
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_2010
0323.007.02">
  <V n="TYPE">Car_Rental</V>
  <V n="STATUS">1</V>
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_TM00
0003">
  <V n="TYPE">Rental_Agreement</V>
  <V n="STATUS">1</V>
  etc.
  </P>
</D>
  etc.
<D
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_2010
0323.007.04">
  <V n="TYPE">Flight</V>
  <V n="STATUS">1</V>
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_TM00
0006">
  <V n="TYPE">Air_Ticket</V>
  <V n="STATUS">1</V>
  etc.
  </P>
</D>
etc.

```

A status of 0 indicates that there are no problems, and a status of 1 indicates that a problem exists. The three problem pages that are shown in the preceding have `Status = 1` for different reasons. To see the nature of the problems, review the individual page files: `tm000001.xml`, `tm000003.xml`, and `tm000006.xml`.

TM000001

The following example shows a portion of the `tm000001.xml` page file:

```

<?xml-stylesheet type="text/xsl"
href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_...\dco.
xsl"?>
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_TM00
0001">
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_Pick
up_Date">
  <V n="TYPE">Pickup_Date</V>
  <V n="Position">189,403,567,465</V>
  <V n="STATUS">0</V>
  <C cn="7" cr="200,416,220,440">84</C>

```

```

    <C cn="4" cr="218,415,226,430">114</C>
    <C cn="10" cr="218,423,230,438">117</C>
    etc.
</F>
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_Pick
up_Location">
  <V n="TYPE">Pickup_Location</V>
  <V n="Position">195,537,558,592</V>
  <V n="STATUS">0</V>
  <C cn="10" cr="203,549,216,570">66</C>
  <C cn="10" cr="219,555,234,570">111</C>
  etc.
</F>
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_Retu
rn_Date">
  <V n="TYPE">Return_Date</V>
  <V n="Position">580,403,942,465</V>
  <V n="STATUS">0</V>
  <C cn="6" cr="593,416,604,438">70</C>
  <C cn="6" cr="606,423,615,438">114</C>
  <C cn="7" cr="619,416,621,438">105</C>
  <C cn="10" cr="625,434,630,441">44</C>
  <C cn="10" cr="690,416,691,438">32</C>
  etc.
</F>
etc.

```

All of the fields in TM000001 have Status = 0 (OK), but the pickup date and return date fields have low confidence characters. By default, any character with a confidence level below 8 is considered low confidence and is displayed to an operator for verification.

TM000003

The following example shows a portion of the tm000003.xml page file:

```

<?xml-stylesheet type="text/xsl"
href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_...\dco.
xsl"?>
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_TM00
0003">
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_Pick
up_Date">
  <V n="TYPE">Pickup_Date</V>
  <V n="Position">0,0,0,0</V>
  <V n="STATUS">0</V>
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_Pick
up_Location">
  <V n="TYPE">Pickup_Location</V>
  <V n="Position">0,0,0,0</V>
  <V n="STATUS">0</V>
</F>

```

Because you only defined recognition zones for the first fingerprint of each page type, TM000003 has no data that is associated with any of the fields. Page TM000003 is the rental agreement page for Car Rental #2 and has no recognition zones. Fix this problem and then run the batch again.

TM000006

The following example shows a portion of the tm000006.xml page file:

```
<?xml-stylesheet type="text/xsl"
href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_...\dco.
xsl"?>
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_TM00
0006">
etc.
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_Airf
are">
  <V n="TYPE">Airfare</V>
  <V n="Position">359,805,527,854</V>
  <V n="STATUS">1</V>
  <V n="MESSAGE">Failed By Calculate Action On Field &apos;TM000006&apos;.</V>
  etc.
</F>
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_Taxe
s">
  <V n="TYPE">Taxes</V>
  <V n="Position">359,861,525,905</V>
  <V n="STATUS">1</V>
  <V n="MESSAGE">Failed By Calculate Action On Field &apos;TM000006&apos;.</V>
  etc.
</F>
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev131_Tota
l_Cost">
  <V n="TYPE">Total_Cost</V>
  <V n="Position">361,912,527,961</V>
  <V n="STATUS">1</V>
  <V n="MESSAGE">Failed By Calculate Action On Field &apos;TM000006&apos;.</V> etc.
</F>
</P>
```

In TM000006, the Calculate('Airfare' + 'Taxes' = Total_Cost) validation action failed. Since Datacap cannot know which of the field values is incorrect, it flags all fields.

Parent topic: [TravelDocs: Update the application to complete validation](#)

Creating recognition zones for the remaining fingerprints

After you review the page status and field status and confirm that the application is working properly, create the recognition zones for the remaining fingerprints.

Refer to [TravelDocs: Specification of recognition zones](#) for instructions on how to create the recognition zones for the different page types.

You need to create recognition zones for each of the following fingerprints.

- Rental_Agreement (Car Rental #2)
- Optional_Insurance (Car Rental #2)
- Rental_Agreement (Car Rental #3)
- Optional_Insurance (Car Rental #3)
- Room_Receipt (Hotel #2)
- Room_Receipt (Hotel #3)
- Air_Ticket (Airline #2)
- Air_Ticket (Airline #3)

Important: As you draw the zones, click Save in the Document Hierarchy pane often.

Drawing the check box recognition zones

To get accurate recognition on the check box options, it is important that all the check box recognition zones on all fingerprints be as close to the same size as possible. You might find it difficult to make the zones the same size when you draw zones on the Zones tab. To establish approximate zone boundaries, draw the bounding boxes on the Image View tab, and then edit the coordinates in the *Pos* variables manually in the Properties pane. For more information, see the section "Implications of using *RecogOMRThreshold*" in the topic [Using the pixel threshold evaluation method](#) for more information.

Parent topic: [TravelDocs: Update the application to complete validation](#)

Running a batch through the workflow

After you define all of the required recognition zones, you can run a batch through the workflow.

Procedure

1. In Datacap Studio, click the Test tab.
2. In the Workflow pane, select the VScan task profile under Main Job.
3. Click New to start a new batch.
4. Click Process rules for target object on the main Test tab toolbar. When you are prompted to release the batch, click Advance. The batch is moved to the next step in the workflow, which is PageID.
5. Click the Process rules for target object button on the main Test tab toolbar. When you are prompted to release the batch, click Advance. The batch is moved to the next step in the workflow, which is Profiler.
6. Click Process rules for target object on the main Test tab toolbar and wait while the task profile runs. When you are prompted to release the batch, click Advance. The batch is moved to the next step in the workflow, which is Verify.
7. Review each of the pages in the Runtime batch hierarchy pane to ensure that recognition was successful. Then, review the batch and page XML files in the runtime batch folder.

Parent topic: [TravelDocs: Update the application to complete validation](#)

Page and field status codes in the TravelDocs application

After you run a batch through the workflow, review the status codes for each of the fields that you validated and for the pages in the runtime batch.

The following table describes how to interpret the status codes.

Field	STATUS = 0	STATUS = 1
Car_Rental		
Rental_Agreement	Page OK	Page contains unrecognized or low confidence characters, or a field with Status = 1
Car_Type	Field OK	Field value is not one of the valid values
Total_Cost	Field OK	Field value is not currency

Field	STATUS = 0	STATUS = 1
Optional_Insurance	Page OK	Page contains unrecognized or low confidence characters, or a field with Status = 1
Total_Cost	Field OK	Field value is not currency
Hotel		
Room_Receipt	Page OK	Page contains unrecognized or low confidence characters, or a field with Status = 1
Total_Cost	Field OK	Field value is not currency
Flight		
Air_Ticket	Page OK	Page contains unrecognized or low confidence characters, or a field with Status = 1
Airfare	Field and all calculated fields OK	Field value is invalid or calculated fields do not add correctly
Taxes	Field and all calculated fields OK	Field value is invalid or calculated fields do not add correctly
Total_Cost	Field and all calculated fields OK	Field value is invalid or calculated fields do not add correctly

Parent topic: [TravelDocs: Update the application to complete validation](#)

Data verification

During verification, Datacap displays pages to an operator for manual checking and possible correction.

There are three primary reasons to display pages to an operator:

- The batch failed document integrity checking.
- A page contains one or more characters or OMR fields that were marked `low confidence` by the recognition engine.
- A page does not pass a validation rule because there is a problem with the integrity of the data.
- [Field data verification](#)
During verification, an operator confirms that data is accurate or, if necessary, corrects problem fields.
- [Skipping a verification task](#)
You can configure your application to skip the verification task and proceed directly to exporting the data when every page passes validation.
- [TravelDocs: Batch verification](#)
You can use Datacap Desktop or the Datacap Web Client for verification.

Parent topic: [Datacap application development](#)

Field data verification

During verification, an operator confirms that data is accurate or, if necessary, corrects problem fields.

Problem fields can include various issues:

- Character fields with one or more low confidence characters

- OMR fields with low confidence values
- Fields with validation errors
- [Options for data verification](#)
Datacap Desktop and Datacap Web Client are two user interface options for verification.
- [Confidence levels and the page status](#)
You can configure your application so that the confidence levels of the fields or characters within a page determine the status for that page.
- [Overriding validation failures](#)
By default, all validations can be overridden, which means that the operator can submit a batch that contains validation errors by selecting to override them

Parent topic: [Data verification](#)

Options for data verification

Datacap Desktop and Datacap Web Client are two user interface options for verification.

Datacap applications can support any or all verification options simultaneously. All verification clients access the same job queue. The clients also provide similar functions, such as identifying and correcting problems, and submitting the batch to the next stage in the workflow.

Datacap Desktop

Datacap Desktop panels are .NET forms. The default field-at-a-time interface is generated automatically from the application's document hierarchy.

You can also create custom panels by using the Datacap Desktop panel builder, which is distributed as a Microsoft Visual Studio project. Custom panels typically display all of a page's fields simultaneously.

Datacap Web Client

Datacap Web Client generates verification panels automatically from the document hierarchy. However, it is also possible to create static layouts and add other custom functions. The web page for the Verifine verification client includes various components:

- An image pane that displays the current page
- A data entry panel that displays image snippets and controls for checking and correcting the data fields
- A batch tree view for restructuring the batch

Datacap Web Client is functionally similar to Datacap Desktop in that the operator must review each problem page, make any necessary corrections, and submit the batch when complete.

Parent topic: [Field data verification](#)

Confidence levels and the page status

You can configure your application so that the confidence levels of the fields or characters within a page determine the status for that page.

- [Confidence levels](#)
During recognition, Datacap assigns a confidence level to each character and OMR field. Confidence levels range from 1 (lowest confidence) to 10 (highest confidence).

- [Page status](#)
The confidence level does not directly affect the page status. For example, a page might have every character with a confidence level of 1 (lowest confidence) but the page status is 0 (good). To set the page status that is based on the confidence level of the characters on the page, use the `ChkConfidence` action.
- [Overriding the default confidence value on specific fields](#)
To determine which characters to display in red, the Datacap Web Client uses a confidence level of 10, unless the field has its own `ReqConf` value.

Parent topic: [Field data verification](#)

Confidence levels

During recognition, Datacap assigns a confidence level to each character and OMR field. Confidence levels range from 1 (lowest confidence) to 10 (highest confidence).

You can see the confidence level for each character or OMR field in the `cn` attribute of the object in the page data file.

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev327_Pick
up_Date">
  <V n="TYPE">Pickup_Date</V>
  <V n="Position">189,403,567,465</V>
  <V n="STATUS">0</V>
  <C cn="7" cr="205,414,219,439">83</C>                                <-- ASCII
'T' [low confidence]
  <C cn="4" cr="205,414,219,439">83</C>                                <-- ASCII
'r' [low confidence]
  <C cn="10" cr="224,423,236,438">117</C>                            <-- ASCII 'u' [high
confidence]
  <C cn="10" cr="241,423,255,438">101</C>                            <-- ASCII 'e' [high
confidence]
  <C cn="10" cr="256,423,266,438">115</C>                            <-- ASCII 's' [high
confidence]
  <C cn="10" cr="270,434,275,441">44</C>                                <-- ASCII
', ' [high confidence]
  <C cn="10" cr="334,416,335,438">32</C>                                <-- ASCII '
' [high confidence]
  <C cn="10" cr="288,416,304,438">68</C>                                <-- ASCII
'D' [high confidence]
  <C cn="10" cr="308,423,320,438">101</C>                            <-- ASCII 'e' [high
confidence]
  etc.
</F>
```

The confidence level determines how Datacap displays the character and the parent field within the verification panel:

- The two verification clients display fields that contain low confidence characters in yellow, where low confidence in this case is anything less than 10.
- Within the field, the Datacap Web Client displays the low confidence characters in red, while Datacap Desktop highlights the problem characters in yellow within the image snippet. Low confidence in this case is anything less than 10 or the field's `ReqConf` value. (See [Overriding the default confidence value on specific fields](#).)

Parent topic: [Confidence levels and the page status](#)

Page status

The confidence level does not directly affect the page status. For example, a page might have every character with a confidence level of 1 (lowest confidence) but the page status is 0 (good). To set the page status that is based on the confidence level of the characters on the page, use the ChkConfidence action.

Library	Action	Description
DCO	ChkConfidence	Checks the confidence level of all characters. If the confidence level on any character is less than the value specified in parameter 1, the action assigns the status value in parameter 2 to the page.

The Datacap Studio application wizard generates a default Routing ruleset the uses this action to set the page status.

Routing Rule 1 is assigned to each page in the document hierarchy and works as follows:

- ChkDCOStatus checks the page status and returns True if the page status is 1. A status of 1 typically signifies that there is an error on the page. If the action returns True, function 2 does not run.
- ChkDCOStatus returns False if the page status is 0 (or any other value other than 1). A status of 0 typically signifies that the page contains no errors. If the action returns False, function 2 runs.
- ChkConfidence examines the characters on the current page and sets the page status to 1 if any character has a confidence level of less than 8 (or the ReqConf value of the field).

Following execution of Routing Rule 1, any page that contains a validation error or a character with a confidence level less than 8 has `Status = 1`. You can configure Datacap to display only pages with `Status = 1` as described in [Show validation failures to an operator](#).

Parent topic: [Confidence levels and the page status](#)

Overriding the default confidence value on specific fields

To determine which characters to display in red, the Datacap Web Client uses a confidence level of 10, unless the field has its own *ReqConf* value.

About this task

Similarly, the ChkConfidence action uses the confidence value that is specified in parameter 1, unless the field has its own *ReqConf* value. For example, if you specify 8 as the parameter value but a field has `ReqConf=6`, ChkConfidence uses the value 6 for that field.

Procedure

To set the confidence level on a specific field:

1. In the Document Hierarchy pane, lock the document hierarchy for editing.
2. Right-click the field and choose Manage variables.
3. If the field has a *ReqConf* variable, assign the appropriate value; otherwise click New, type `ReqConf`, press Enter, and then assign the value.
4. Click Done and then click Save in the Document Hierarchy pane.

Overriding validation failures

By default, all validations can be overridden, which means that the operator can submit a batch that contains validation errors by selecting to override them

Depending on the business requirements, overriding validation failures is appropriate. For example, if the validation error stems from a calculation error on the original page, the operator does not modify the field values. Instead, the operator must override the error and submit the batch. The application sends the batch to an exception handling task. When the operator overrides a validation error, Datacap sets the page status to 73 and the document status to 142.

In other situations, you can prevent the operator from overriding validation errors by using the `SetIsOverrideable` action.

Library	Action	Description
Validations	SetIsOverrideable	If set to <code>False</code> , specifies that if validation on the current object fails, the operator cannot override the error. If set to <code>True</code> , the operator can override the error.

For example, to prevent the operator from overriding an error in the Validate Car Type rule, you can insert `SetIsOverrideable("False")`.

```
Validate Car Type
  Validation: Car Type
    SetIsOverrideable("False")
    OpenConnection("@APPVAR(*.lookupdb:cs)")
    ExecuteSQL("""SELECT Car_Type FROM Car_Types WHERE Car_Type='%s';",Car_Type")
    CloseConnection()
```

In this case, the operator must select a valid car type from a drop-down menu that Datacap Desktop displays next to the field that failed validation.

If the operator attempts to submit a page that failed this validation, an error message is shown.

Parent topic: [Field data verification](#)

Skipping a verification task

You can configure your application to skip the verification task and proceed directly to exporting the data when every page passes validation.

About this task

You need to set the mode to Router for the task that precedes the task that you want to skip. In this case, the Profiler task precedes the verification task. Also, you must create a ruleset that includes a condition to cause the skip. Finally, you must add the ruleset at the end of the Profiler task in Datacap Studio.

Procedure

1. Start Datacap Web Client and log in to the application that contains that task to be skipped.

2. Click the Administrator, expand the job that contains the verification task, and select the task before the verification task. In most applications, the Profiler task (the task that applies validation rules) precedes the verification.
3. In the Selected task details section, select Router from the menu for the Mode field.
4. Under Parameters, enter a value, such as `Skip` for the Returned Conditions key.
5. Select the condition (Skip), and in the Selected condition details window, select Jump from the menu for the Spawm type field, and enter `1` for the Steps field. A value of `1` configures the workflow so that one task is skipped when the condition is encountered. In this case, the condition is every page passed validation.
6. Start Datacap Studio and log in to the same application that you configured in Datacap Web Client in steps 1 - 5.
7. Click the Rulesets tab, lock the rulesets for editing, and add a ruleset with a name such as `Skip Verify`.
8. Rename `Rule 1` to a name such as `Check Confidence`.
9. Click `Function1`, select an action from the corresponding action library, and click `Add to Function` to add the action to `Function1`. Refer to the table to determine the action library from which to select an action and the parameter values to enter.

Important: You must add the actions in the order shown.

Action Library	Action	Values to enter	Description
DCO	SetPageStatus	0	0 indicates that the page passed validation.
DCO	ChkConfidence	8, 1	The first parameter sets a high-confidence level, and 1 is the code to return for low confidence fields on the page.
RRunner	Task_NumberOfSplits	1	1 indicates that there is one subbatch.
RRunner	Task_RaiseCondition	0, 0	The first value in the pair indicates the subbatch index, where 0 is the first subbatch. The second value raises the first Child Job Condition (in this case, Skip Verify) for the subbatch entry.

10. Save and publish the ruleset.
11. Click the Task Profiles tab, select the Profiler task profile, and lock it for editing.
12. In the Rulemanager tab, select Skip Verify (or the ruleset that you created in step 7), and click `Add ruleset to profile` to add the ruleset to the Profiler task profile.
13. Click `Save` in the Task Profiles tab, and unlock the Profiler task profile.
14. Add this new ruleset to the DCO page object.

Parent topic: [Data verification](#)

TravelDocs: Batch verification

You can use Datacap Desktop or the Datacap Web Client for verification.

- [Setting the Car Type field to prevent overriding](#)
Before you run a batch through the workflow, set the `SetIsOverrideable` property on the Car Type field to `False` to prevent the operator from overriding an invalid car type.
- [Batch verification with Datacap Desktop](#)
Datacap Desktop is a thick client that you can customize and use for verification.

- [Verifying batches with Datacap Web Client](#)

Before you can use Datacap Web Client for verification, you must configure the Datacap Web Client server.

Parent topic: [Data verification](#)

Setting the Car Type field to prevent overriding

Before you run a batch through the workflow, set the `SetIsOverrideable` property on the Car Type field to `False` to prevent the operator from overriding an invalid car type.

Procedure

1. On the Datacap Studio Rulemanager tab, Rulesets pane, select the Validate ruleset and click Lock/Unlock ruleset. The ruleset is locked for editing.
2. Expand the Validate Car Type rule completely.
3. Select the Validation: Car Type function.
4. Expand the Validations library and select `SetIsOverrideable`.
5. Click Add to function at the left side of the Actions Library pane. The action is added to the Validate ruleset.
6. In the Properties pane, set `StrParam` to `False`.
7. Use the Up Arrow button to move the new action to the beginning of the function.
8. In the Rulesets pane, click Save, click Lock/Unlock ruleset, and choose Publish Ruleset.

Parent topic: [TravelDocs: Batch verification](#)

Batch verification with Datacap Desktop

Datacap Desktop is a thick client that you can customize and use for verification.

- [Creating dictionaries for check box options](#)
You can use Datacap Desktop for verification, but you must create dictionaries for the check box options, and attach the dictionaries to the check box fields.
- [Preparing a batch for verification](#)
For demonstration purposes, run a batch in the Datacap Web Client through the Profiler task so that the batch is pending verification.
- [Opening the batch in Datacap Desktop](#)
Use Datacap Desktop to open batches that are pending verification.
- [Reviewing the batch in Datacap Desktop](#)
When you start Datacap Desktop and open a batch that is pending verification, Datacap Desktop defaults to the field-at-a-time interface and displays the first problem field.
- [Submitting the batch](#)
After you review each problem page in a batch and make any necessary corrections, you can submit the batch.

Parent topic: [TravelDocs: Batch verification](#)

Creating dictionaries for check box options

You can use Datacap Desktop for verification, but you must create dictionaries for the check box options, and attach the dictionaries to the check box fields.

Procedure

1. Create the dictionaries for the check box options.
 - a. On the Datacap Studio Rulemanager tab, Document Hierarchy pane, click Lock DCO to lock the document hierarchy for editing.
 - b. Click Dictionaries in the Document Hierarchy pane.
 - c. Click Edit dictionary and select Add dictionary.
 - d. Change the dictionary name from `<new_dictionary>` to `Options`.
 - e. Right-click the new dictionary and choose Add word.
 - f. Change the name from `<new word>` to `Navigation System` and the value from `value` to `Navigation System`.
 - g. Repeat to add `Child Seat` and `Fuel Service` to the dictionary.
 - h. Click Edit dictionary and select Add dictionary.
 - i. Create another dictionary that is called `Checkbox` and add one word with the name `Selected` and a value `Selected`.
 - j. Click Save in the dialog, and then click Save in the Document Hierarchy pane.
2. Attach the dictionaries to the check box fields.
 - a. Confirm that the document hierarchy is still locked for editing.
 - b. Expand the `Car_Rental > Rental_Agreement` page so that the fields are visible.
 - c. Right-click the `Options` field and choose Manage variables.
 - d. Click New, enter `DICT`, and press the Enter key.
 - e. Enter the value `Options`, and click Done.
 - f. Expand the `Car_Rental > Optional_Insurance` page so that the fields are visible.
 - g. Right-click the `CDW` field and choose Manage variables.
 - h. Click New, enter `DICT`, and press the Enter key.
 - i. Enter the value `Checkbox`, and click Done.
 - j. Repeat for the `PAI`, `PEP`, and `ELP` fields.
 - k. In the Document Hierarchy pane, click Save, and then click Unlock DCO.

Parent topic: [Batch verification with Datacap Desktop](#)

Preparing a batch for verification

For demonstration purposes, run a batch in the Datacap Web Client through the Profiler task so that the batch is pending verification.

Procedure

To prepare a batch for verification:

1. Start the Datacap Web Client (if it is not already running).
 - a. Select the TravelDocs application.
 - b. Log in using `User ID: admin`, `Password: admin`, and `Station: 1`.
2. Select the Administrator tab and then click Operations.
3. Click VScan and wait for the task to complete. Then, click Stop.
4. On the Operations page, click PageID and wait for the task to complete. Then, click Stop.
5. On the Operations page, click Profiler and wait for the task to complete. Then, click Stop. The batch is now pending verification.

Parent topic: [Batch verification with Datacap Desktop](#)

Opening the batch in Datacap Desktop

Use Datacap Desktop to open batches that are pending verification.

Procedure

To open the batch in Datacap Desktop:

1. In the Start menu click IBM Datacap Clients Datacap Desktop.
2. Enter these values in the login panel.
 - o Application: TravelDocs
 - o User ID: admin
 - o Password: admin
 - o Station: 1
3. Click Login.
4. In the Shortcut field, select Verify, and click Start.
 - o If there are no older batches on hold, click OK to run the next pending batch.
 - o If there are older batches on hold, click Run Pending! to run the pending batch.

Parent topic: [Batch verification with Datacap Desktop](#)

Reviewing the batch in Datacap Desktop

When you start Datacap Desktop and open a batch that is pending verification, Datacap Desktop defaults to the field-at-a-time interface and displays the first problem field.

Within Datacap Desktop, you can go directly to the next problem by clicking Next Problem.

You can display any page in the batch by selecting it in the Batch View pane.

You cannot modify the default field-at-a-time interface. Instead, if you want to modify the Datacap Desktop interface you must create a custom panel for each page.

Parent topic: [Batch verification with Datacap Desktop](#)

Submitting the batch

After you review each problem page in a batch and make any necessary corrections, you can submit the batch.

Procedure

To submit the batch:

1. Review each problem page and complete these steps.
 - a. Correct any low confidence fields.
 - b. Correct the Car Type validation failure on the third car rental page by selecting Other.
 - c. Ignore the other fields with validation failures because the errors are on the original images.
2. Click Submit advance to the next problem page. When you are prompted to the override validation failures, click OK.
3. When you reach the end, click OK to finish the batch. The batch is now marked as pending for export.
4. Close Datacap Desktop.

Parent topic: [Batch verification with Datacap Desktop](#)

Verifying batches with Datacap Web Client

Before you can use Datacap Web Client for verification, you must configure the Datacap Web Client server.

About this task

This task provides an outline of how to set up the Datacap Web Client server by using Microsoft Information Systems (IIS) 7.5 and Microsoft Windows 7 Professional.

By default the Datacap Web Client loads only the data file for the current image when a Verify task is run. You cannot run cross document or cross batch validations. To be able to run validation on an entire batch or a document, you can configure the caching option by setting the LoadDoc value in the <verify>.set.xml file to load all of the data files in the current batch or only for current document.

Procedure

Follow this procedure to use the Datacap Web Client for verification.

1. Set up the Datacap Web Client server:
 - a. In the Start menu, select IBM Datacap Web > Datacap Web Server Configuration Tool. The message box notifies you that the IIS components are installed.

```
Internet Information Services (IIS) 7.5:Found
IIS Component - IIS Management Console:Found
IIS Component - ASP.NET:Found
IIS Component - ASP:Found
IIS Component - Static Content:Found
```
 - b. If the message indicates that you are missing components, open the Programs and Features window from Windows Control Panel, click Turn Windows features on or off, and install the missing components.
 - The Management Console is under Internet Information Services > Web Management Tools.
 - The ASP and ASP.NET components are located under Internet Information Services > World Wide Web Services > Application Development Features.
 - The Static Content component is located under Internet Information Services > World Wide Web Services > Common HTTP Features.
 - c. After you install the missing components, run the Datacap Web Client Server Configuration tool again. Then, click OK to continue.
 - d. In the Datacap Web Client Server Configuration window, click Configure to set up the Datacap Web Client application in IIS.
 - e. Start the Internet Information Services (IIS) Manager or type `inetmgr`.
 - f. Expand Sites > Default Web Site and confirm that the TaskRun folder shortcut and the tmweb.net application are visible.
2. Set up the Datacap Web Client:
 - a. In the Start menu click IBM Datacap Web Datacap Web Server Configuration Tools
 - b. Click Configure to set up the security options in Internet Explorer.
3. Configure the caching option in the <verify>.set.xml file:
 - a. Go to C:\Datacap\ProgramSet and open the <verify>.set.xml for your verification task.
 - b. Click Search > Find and enter `LoadDoc`.
 - c. Set the LoadDoc checkbox parameter by using one of the following options:
 - 0: cache only the current data file
 - 1: cache all of the data files in the current document
 - 2: cache all of the data files in the current batch
4. Create dictionaries for the check box options:
 - a. If you did not complete the section on using Datacap Desktop for verification, complete the procedures in [Creating dictionaries for check box options](#) to create and attach the dictionaries to

- the check box fields.
5. Prepare a batch for verification:
 - a. Start the Datacap Web Client, if it is not already running.
 - Select the TravelDocs application and click OK.
 - Log in using User ID: admin, Password: admin, and Station: 1.
 - b. Select the Administrator tab and then click Operations.
 - c. Click VScan and wait for the task to complete. Then, click Stop.
 - d. On the Operations page, click PageID and wait for the task to complete. Then, click Stop.
 - e. On the Operations page, click Profiler and wait for the task to complete. Then, click Stop. The batch is now pending verification.
 6. On the Operations page, click Verify.
 7. Review and submit the batch: Datacap Web Client displays the page with the fields and image snippets automatically.
 - a. Review each problem page and complete these steps:
 - Correct any low confidence fields.
 - Correct the Car Type validation failure on the third car rental page by selecting Other.
 - Do not correct the other fields with validation failures because the errors are on the original images.
 - b. Click Submit at the top of the verification panel to advance to the next problem page. When you are prompted to the override validation failures, click OK
Tip: If you are unable to get past the page with the validation error after you click OK, click Hold to put the batch on hold. Then, click the Administrator tab, select the Verify task in the Workflow page, and click Setup in the Selected task details pane. In the Webpage Dialog Navigation panel, enter a value of 0, 2 for the Done Page Status field.
After you complete the revision and save the setup file, reopen the batch from the Datacap Web Client Monitor tab by clicking the QID field. For information about the Done Page Status and other settings, see [Configuring the VeriFine client](#).
 - c. When you reach the end of the batch, click OK to finish the batch. The batch is now marked as pending for export.

For more information about configuring the web verification client, see [Verification by using the VeriFine web client](#).

Parent topic: [TravelDocs: Batch verification](#)

Related information:

[Setting Datacap Navigator default page layouts](#)

Data export

Datacap can export data to a text file, an XML file, a database, a Document Management system, or a custom business process. The default output format is a text file, but you can use some actions to export data to a database and an XML file.

- [Exporting data](#)
You can export data to different file types or to specific repositories. To export to a Content Manager OnDemand repository, you must consider other configuration options. Also, you can use Datacap connector actions to automate capture and indexing processes.
- [TravelDocs: Exporting data to a database](#)
You can update the TravelDocs application to export data from each rental agreement page to an export database.
- [TravelDocs: Exporting data to an XML file](#)
You can update the TravelDocs application to export data from each rental agreement page to an XML file. If you want to export data from the other pages, you must have a separate rule for each page type.

Exporting data

You can export data to different file types or to specific repositories. To export to a Content Manager OnDemand repository, you must consider other configuration options. Also, you can use Datacap connector actions to automate capture and indexing processes.

- [Export to a text file](#)
The Application Wizard generates a framework that exports all captured data to a text file in the application's export folder.
- [Configure text export for IBM Content Manager OnDemand](#)
You can configure Datacap to export index data and files into Content Manager OnDemand.
- [Export to a database](#)
Datacap can export data to any DB2®, Microsoft Access, Microsoft SQL Server, or Oracle database by using the actions in the ExportDB library
- [Export to an XML file](#)
Datacap can export data to an XML file by using the actions in the ExportXML action library.
- [Datacap Connector actions](#)
Datacap provides actions that you can use to integrate Datacap applications with supported content repositories. You can automate data capture, index documents, and process forms as a front end that stores document images and associated index values in the repository.

Parent topic: [Data export](#)

Export to a text file

The Application Wizard generates a framework that exports all captured data to a text file in the application's export folder.

The default Export ruleset includes two rules:

- **Set Export Params:** This rule is attached to the batch's `Open` element. It sets the export path and file name, and writes the header information to the file.
- **Export Page Fields:** This rule is attached to each page and writes all fields values from the current page to the export file.

The resulting file looks like the following example, where each line represents one page and the fields are separated by commas.

```
*****
Export for batch #20100334.019,12/01/2010,08:47:58 <--- Header information
,Tues, Dec 7, 2010,Boston (BOS),Fri, Dec 10, 2010,Boston (BOS),Compact,001,$345.70
,0,0,0,1
,Mon, Dec 6, 2010,San Francisco (SFO),Fri, Dec 10, 2010,San Francisco
(SFO),SUV,010,$489.31
,Boston (BOS),Pittsburgh (PIT),17NOV10,Pittsburgh (PIT),Boston
(BOS),21NOV10,313.17,64.56,477.73
,Newark, NJ (EWR),Charlotte, NC (CLT),MON NOV 15, 2010,Charlotte, NC (CLT),Newark,
NJ (EWR),WED NOV 17, 2010,$524.76,$53.23,$577.99
,Dec 21, 2010,Dec 24, 2010,$293.03
,Nov 30, 2010,Dec 2, 2010,$243.07
```

The Export actions library includes actions that are typically used for exporting captured data to a text file. A few of the key export actions are outlined in the following table.

Library	Action	Description
Export	SetExportPath	Specifies the path to the export file's location. Typically you reference the export path in the application's configuration file by using the @APPPATH(export) smart parameter.
Export	SetFileName	Specifies the name for the export file (do not include the file extension).
Export	SetExtensionName	Specifies the extension for the export file.
Export	SetExportFileEncodingAsASCII	Specifies whether to export files in ANSI format. If you do not add this action, files are exported in Unicode format by default.
Export	ExportAllFields	Writes all field values on the current page to the export file.
Export	ExportFieldValue	Writes the specified field's value to the export file, for example, ExportFieldValue(Return_Date) .
Export	CloseExportFile	Closes the export file.

Parent topic: [Exporting data](#)

Configure text export for IBM® Content Manager OnDemand

You can configure Datacap to export index data and files into Content Manager OnDemand.

The Content Manager OnDemand tool contains the ARSLOAD component. ARSLOAD can enter a flat index file that contains index data and locations of files that can be uploaded with the index data. You can use the generic Export library action to create output index files in a format that is required by ARSLOAD.

For detailed information on ARSLOAD file formats and how to configure your Datacap system to export data into Content Manager OnDemand, see <http://www-01.ibm.com/support/docview.wss?uid=swg21502807>.

Parent topic: [Exporting data](#)

Export to a database

Datacap can export data to any DB2®, Microsoft Access, Microsoft SQL Server, or Oracle database by using the actions in the ExportDB library

Commonly used export actions are outlined in the following table.

Library	Action	Description
ExportDB	ExportOpenConnection	Opens a connection to the specified export database.
ExportDB	SetTableName	Specifies the name of the table to which data is to be exported.
ExportDB	ExportFieldToColumn	Gets the value of the specified field on the current page and adds it to specified column in the internal data record. You build the record in memory before you commit it to the database by using AddRecord.
ExportDB	AddRecord	Inserts the assembled data record into the export table that is specified by the previous SetTableName action.
ExportDB	ExportCloseConnection	Closes an open export database connection.

For a complete example, see [Creating the ExportDB ruleset](#).

Parent topic: [Exporting data](#)

Export to an XML file

Datacap can export data to an XML file by using the actions in the ExportXML action library.

The ExportXML library includes actions that you can use to write data to an XML file. Some of the export actions are described in the following table.

Library	Action	Description
ExportXML	xml_SetExportPath	Specifies the path to the XML file storage location.
ExportXML	xml_SetFileName	Specifies the name for the XML file (do not include the .xml extension).
ExportXML	xml_NewNode	Creates a child node under the specified parent node, creating the parent node if necessary.
ExportXML	xml_SetFileEncodingAsASCII	Specifies whether to export files in ANSI format. If you do not add this action, files are exported in Unicode format by default.
ExportXML	xml_SetNodeValue	Sets the value of the specified node.

Library	Action	Description
ExportXML	xml_SaveFile	Commits all unsaved nodes and saves the XML file to disk.

For a complete example, see [Creating the ExportXML ruleset](#).

Parent topic: [Exporting data](#)

Datacap Connector actions

Datacap provides actions that you can use to integrate Datacap applications with supported content repositories. You can automate data capture, index documents, and process forms as a front end that stores document images and associated index values in the repository.

Datacap Connector actions are associated with objects in the Document Hierarchy at the batch, document, page, or field level through rulesets. For example, to export objects from a Datacap application to IBM® Content Manager. You configure rules on the IBMCM Export ruleset to upload Datacap scanned images to IBM Content Manager.

You can use the IMail or EWSMail input actions to scan incoming email messages for attachments that can then be imported into document batches. You can use the Email output actions to send email messages. The Fax actions can import the content of incoming faxes from a specified fax source into document batches. You can process email and fax-based batches by using the Datacap standard Recognition or Verify tasks.

You configure these actions in a Datacap 8.0 or later Windows environment.

- [Verifying the installation](#)
Before you start to configure these actions, verify the Datacap Connector actions were installed as part of a complete Datacap product installation.
- [Content repository authentication](#)
You must have write access to a folder on the repository and privileges to create and view documents in that folder. Then, you can use the Datacap Connector Actions to upload documents into the repository.
- [Integrating Connector actions into applications](#)
To integrate Datacap Connector actions into Datacap applications, you add the actions that you want to use to functions in your rulesets.
- [Connector actions configuration](#)
To export documents and index files into Content repositories and libraries, you must add the Connector actions to the appropriate Export rulesets.
- [IBM Content Manager Connector actions](#)
The IBM Content Manager Connector actions integrate Datacap applications with the IBM Content Manager repository.
- [FileNet P8 Connector actions](#)
The FileNet® P8 Connector actions integrate Datacap applications with IBM FileNet Content Engine.
- [Documentum Connector actions](#)
The Documentum Connector actions integrate Datacap applications with a Documentum Docbase content repository.
- [SharePoint Connector actions](#)
The Datacap Connector for Microsoft SharePoint actions integrate Datacap applications with Microsoft Office SharePoint Services (MOSS) for Microsoft SharePoint 2007 and 2010.
- [FileNet Image Services Connector Connecting actions](#)
You use Datacap Connector for FileNet Image Services actions to upload documents and commit images to an IBM FileNet Image Services library.

- [Email Connector actions](#)
Email Connector actions create Datacap batches from the documents that you receive as email attachments. You can also send email notification messages when specific events occur.
- [Fax Connector actions](#)
You can use Fax Connector actions to create Datacap document batches from incoming faxes. You can also send the contents of a document to a specified fax number.
- [Box Connector actions](#)
You can use the DatacapBOX connector actions to move data between your IBM Datacap system and Box.com.
- [Connector actions log files](#)
A log file contains the results of calling the action and explains why a document was not created when you upload documents into a repository. The name of the log file is based on the name of the task, for example export_rss.log.
- [Viewing action details](#)
Datacap Studio provides help topics with detailed information for all of the connector actions. The topics include the action library name, description, parameters, DCO level, returns, and examples for each action.

Parent topic: [Exporting data](#)

Verifying the installation

Before you start to configure these actions, verify the Datacap Connector actions were installed as part of a complete Datacap product installation.

Procedure

To verify the installation:

1. In Datacap Studio, click the Rulemanager tab.
2. Select the Actions library tab.
3. Scroll to Global Actions and verify that the actions file for your repository is listed.
 - For IBM® FileNet® Content Engine, look for FileNetP8
 - For IBM Content Manager, look for Datacap.Libraries.IBMCM
 - For SharePoint, look for SPEXport
 - For IBM FileNet Image Services, look for FileNetIDM
 - For eMail and eDoc Connector, look for IMail, EWSMail, and EMail
 - For Fax Connector, look for OpenTextFaxServer
 - For Box, look for DatacapBOX. Some configuration might be required to make the DatacapBOX action library visible in the user interface of Datacap programs. For more information, see [Configuring Box Connector actions](#).

The name of the action file might not match the name of the *connector_name.rxx* file. If the action file is not listed, the actions must be installed before you can continue.

Parent topic: [Datacap Connector actions](#)

Content repository authentication

You must have write access to a folder on the repository and privileges to create and view documents in that folder. Then, you can use the Datacap Connector Actions to upload documents into the repository.

Access control is handled differently by each of the repositories and their connectors:

- For Datacap Connector for IBM® Content Manager, access is controlled through the IBM Content Manager authentication.
- For Datacap Connector for FileNet® Content Manager, access is controlled through the IBM FileNet Content Manager authentication.
- For Documentum Connector, authentication is done by using the Login action with user credentials that are managed by Documentum.
- For Datacap Connector for Microsoft SharePoint, authentication is done by using the Login action with user credentials that are managed by SharePoint.
- For Datacap Connector for FileNet Image Services, authentication is done by the library into which you are importing the documents.
- For Datacap Connector for eMail and Electronic Documents, authentication is done by the Microsoft Exchange Server from which you are importing the attachments.
- For Datacap Connector for Fax, authentication is done by the Fax Server from which you are importing the faxes.
- For Datacap Connector for Box, authentication is done with your configured OAuth parameter values. For more information, see [Configuring Box Connector actions](#).

For more information about rule sets and tasks, see [Task profiles and rulesets](#).

Parent topic: [Datacap Connector actions](#)

Integrating Connector actions into applications

To integrate Datacap Connector actions into Datacap applications, you add the actions that you want to use to functions in your rulesets.

About this task

The Datacap applications are unique, so you can do some general steps to incorporate connector actions into an application. You must connect the rules to the appropriate levels in the DCO.

Procedure

To integrate connector actions into an application:

1. In Datacap Studio, click the Rulemanager > Actions tab.
2. Select the ruleset to which you want to add the connector action and click Lock ruleset for editing. For example, you might select the Export To P8 ruleset.
3. Click Sync DCO view with Ruleset view to expand the Document Hierarchy.
4. Highlight the objects to which the ruleset is bound and note the object names and their object levels, such as Connect or Upload.
5. Select the function into which you want to incorporate the connector action in the Rulesets tab. For example, select Logon.
6. Select the Page or Fieldlevel action on the Actions Library tab and click Add to Function to add the action to the function. If you selected the Logon function, add the Logon action for your content repository. For example, for IBM® Content Manager, you add the IBMCM_Logon action.
7. If needed, move the action by clicking Move Up or Move Down then change the action Properties as needed.
8. Click Save to save the changes to the ruleset.
9. Display the Connector Settings by clicking the Zones tab, then clicking the Connector tab.
10. On the Document Hierarchy tab, click Lock DCO for editing and select the objects that you highlighted in a previous step.
11. Change the appropriate Connector settings for the selected objects on the Connector tab,

12. On the Document Hierarchy tab, click Save Changes and then click Unlock DCO.

13. Test your changes, then click the Rulesets tab and click Publish ruleset.

- [Storing passwords in the .app file](#)

To pass passwords as action parameters, use smart parameters that retrieve credentials from the .app file where the passwords are stored as encoded strings.

Parent topic: [Datacap Connector actions](#)

Storing passwords in the .app file

To pass passwords as action parameters, use smart parameters that retrieve credentials from the .app file where the passwords are stored as encoded strings.

About this task

You can use smart parameters in a key path to access the passwords for the Datacap Connector actions. See [Reference passwords, connection strings, and other parameters from your actions](#) for information about storing action parameters in the .app file.

Procedure

To store passwords in the .app file:

1. In the Start menu click IBM Datacap Services Datacap Application Manager.
2. Click the Custom values tab and select your application from the list in the left pane.
3. Under the Advanced values field, press Add new.
4. Enter the password name in the Value name field. Create a logical password name for your system, such as *FileNet P8 password*.
5. Enter the password in the Value field.
6. Close the Application Manager.
7. Access the password in the action by using the key path for the password, `@APPVAR(values/adv/<value name>)`. For example, if the value name of the password is FileNet P8 password, the key path is `@APPVAR(values/adv/<FileNet P8 password>)`.

Parent topic: [Integrating Connector actions into applications](#)

Related information:

[Application Manager](#)

Connector actions configuration

To export documents and index files into Content repositories and libraries, you must add the Connector actions to the appropriate Export rulesets.

Open Datacap Studio and use the Export ruleset for the repository or library into which you want to export documents. For example, to export documents into an IBM® Content Manager, you might use a ruleset named Export To CM. You configure the ruleset with rules that log on to Content Manager and upload a document into the repository. These rules might be named Connect to CM and AddDocument.

You then add functions like Login and AddPage to these rules. You then configure the functions with IBM Content Manager Actions that define how to Connect To CM and Add a Document.

Parent topic: [Datacap Connector actions](#)

IBM Content Manager Connector actions

The IBM® Content Manager Connector actions integrate Datacap applications with the IBM Content Manager repository.

Use these actions to upload documents and index fields into an IBM Content Manager repository.

You can configure the IBM Content Manager Connector actions for the following tasks:

- Log in to the IBM Content Manager server
- Search for an item in the IBM Content Manager repository based on an attribute and value or item ID that is provided.
- Create a IBM Content Manager document that is based on the type in the Document Hierarchy as a document or a page
- Add, delete, or replace pages in the IBM Content Manager document as needed
- Set the attribute value on the IBM Content Manager document
- Set the MIME type for the IBM Content Manager document that you are uploading
- Create an IBM Content Manager folder in the parent folder where you can upload documents
- Set the attribute value on the IBM Content Manager folder
- Set the path to the IBM Content Manager folder where you are uploading documents
- Upload the document, page, or directory to the IBM Content Manager server
- Store the item ID of the uploaded IBM Content Manager document or page in the DCO
- Store the ID of the most recently created IBM Content Manager folder into a variable of the Document Hierarchy
- Search and download the existing IBM Content Manager documents
- After the files are downloaded, you can process them in Datacap application to find any additional data in the content

- [IBM Content Manager Connector prerequisites](#)
To configure and run IBM Content Manager Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 9.0 or later.
- [IBM Content Manager Connector settings](#)
Record the system settings that you want to use to configure the IBM Content Manager Connector actions and have these values available during the configuration process.
- [Configuring IBM Content Manager Connector actions](#)
You must create an Export ruleset and configure its rules and functions with IBM Content Manager Connector actions to upload documents from Datacap applications into IBM Content Manager.
- [IBM Content Manager Connector upload examples](#)
The Datacap Connector for IBM Content Manager Upload actions configure the connection between the Datacap application and the IBM Content Manager repository.
- [Search and download action attributes of IBM Content Manager](#)
The Datacap Connector for IBM Content Manager actions can help you to search and download the existing file content.
- [Updating IBM Content Manager content with search and download actions](#)
Processes the downloaded files (using the Search and Download action) in Datacap application to find any additional data in the content.

Parent topic: [Datacap Connector actions](#)

IBM Content Manager Connector prerequisites

To configure and run IBM® Content Manager Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 9.0 or later.

A complete and up-to-date list of the hardware and software requirements for IBM Datacap is available on the IBM support site at <http://www.ibm.com/support/docview.wss?uid=swg27043811>.

The following components must be installed and running on your system before you can use IBM Content Manager Connector actions to upload images into a IBM Content Manager repository.

- Datacap Version 9.0 or later installed and running on either a single computer or a client/server installation
- Network access to a supported version of IBM Content Manager

Supported versions of the following repository clients must be installed on each Datacap computer that runs the export ruleset. Export actions are run on Rulerunner in production. Export actions can also be run in Datacap Studio or Datacap Desktop for development or test purposes. Computers that run rules must have the appropriate clients, such as Rulerunner and Datacap Studio, installed on them.

- IBM DB2® Client
- IBM Content Manager Enterprise Edition
- IBM Information Integrator for Content

Parent topic: [IBM Content Manager Connector actions](#)

Related information:

[Hardware and Software Requirements for IBM Datacap Version 9.0.1](#)

IBM Content Manager Connector settings

Record the system settings that you want to use to configure the IBM® Content Manager Connector actions and have these values available during the configuration process.

This table describes the parameters that are required for Datacap Connector for IBM Content Manager actions.

Table 1. Required IBM Content Manager parameter settings

Action	Description
Logon	Server name, user ID, password
Search Item	The attribute name and value or the item ID of the item for which you want to search in IBM Content Manager. The item that is found is set as the current item for the actions that follow this action in the application.
Create Item	The IBM Content Manager item type, such as document or page.
New page	The pages to add to the existing IBM Content Manager document
Existing page	The existing IBM Content Manager page to delete or replace.
Set Attribute Value	A valid IBM Content Manager item type equivalent to a Document Class such as NOINDEX or a predefined Smart Parameter that contains a valid item type.
Create Folder	An IBM Content Manager folder in the parent folder. This new folder is based on the item type and the parent folder ID.
Create Folder Attribute Value	The attribute name and value of the IBM Content Manager folder or a predefined Smart Parameter that contains a valid attribute name and value.

Action	Description
Set Destination Folder	A valid IBM Content Manager destination folder ID based on the parent folder ID.
Upload Document	None
Upload Page	None
Store Item In DCO	The item ID of the document or page that you are uploading.
Store Folder ID In DCO	The folder ID of the most recently created IBM Content Manager ID.
Search and Download	This action helps to search and download content from IBM Content Manager repository. For search criteria and the download directory, see the topic Search and download action attributes of IBM Content Manager .

Parent topic: [IBM Content Manager Connector actions](#)

Configuring IBM Content Manager Connector actions

You must create an Export ruleset and configure its rules and functions with IBM® Content Manager Connector actions to upload documents from Datacap applications into IBM Content Manager.

Procedure

To configure IBM Content Manager Connector actions:

1. Install the IBM Content Manager Runtime Environment. For more information, see the IBM Content Manager client installation instructions.
2. Restart the Datacap Client Station.
3. Add the IBM Content Manager Connector actions (IBMCM.RXX) to the Export rulesets.

Example

The following example describes an Export To Content Manager ruleset that logs on to the IBM Content Manager server and uploads a single page document into IBM Content Manager.

This ruleset contains the Connect and Upload rules. The Connect rule contains the Logon function and action. The Upload rule contains the AddPage function with actions that create the page, set attribute values for the page, upload, and store the page.

Export To Content Manager ruleset

- Connect rule
 - Logon function
 - IBMCM_Logon("ibmcm srv,userid,password")
- Upload rule
 - AddPage function
 - IBMCM_CreateItem("APT")
 - IBMCM_SetAttributeValue("APT_Title,Page from CM8ItemDCO")

- IBMCM_SetAttributeValue("APT_Date,@P.VerifyTime")
- IBMCM_SetAttributeValue("APT_Vendor,@P.Vendor")
- IBMCM_CreateFolder("APT_Folder", "123456789")
- IBMCM_SetFolderAttributeValue("Name", "APT_Folder")
- IBMCM_SetDestinationFolder("123456789")
- IBMCM_UploadDCO_Page()
- IBMCM_StoreItemIDinDCO("CM8ItemDCO")
- IBMCM_StoreFolderIDinDCO("APT_Folder")

Parent topic: [IBM Content Manager Connector actions](#)

IBM Content Manager Connector upload examples

The Datacap Connector for IBM® Content Manager Upload actions configure the connection between the Datacap application and the IBM Content Manager repository.

You use these actions to upload a single page file or an image that contains multiple pages and their associated index values from Datacap into IBM Content Manager.

These actions are based on the IBM Content Manager Java™ API. If you use the IBM Content Manager Java APIs, you must install the IBM Information Integrator for Content connector for Content Manager on the computers where you want to run these actions.

The examples in the following tables show the sequence in which you must add the actions to the Export To Content Manager ruleset for the upload scenarios.

Upload a single page file

Table 1. The sequence of actions for uploading a single page file into IBM Content Manager

Action	Description
IBMCM_Logon("ibmcmsrv,userid,password")	Log the application on to the IBM Content Manager server.
IBMCM_CreateItem("NOINDEX")	Create an IBM Content Manager document.
IBMCM_SetAttributeValue("USERID,@OPERATOR")	Set an attribute value on the IBM Content Manager document.
IBMCM_SetMimeType("application/msword")	Set the MIME type for the IBM Content Manager document you are uploading.
IBMCM_CreateFolder("NOINDEX", "123456789")	Create an IBM Content Manager folder that is based on the item type and parent folder ID.
IBMCM_SetFolderAttributeValue("Name", "MyFolder")	Set an attribute value on the IBM Content Manager folder.
IBMCM_SetDestinationFolder("\APT")	Identify the folder into which the uploaded is IBM Content Manager.
IBMCM_UploadDCO_Page()	Upload the images that are associated with the current Page object of the document hierarchy to IBM Content Manager.

Upload a multiple page file

Table 2. The sequence of actions for uploading a multiple page file into IBM Content Manager

Action	Description
IBMCM_Logon("ibmcmsrv,userid,password")	Log the application on to the IBM Content Manager server.
IBMCM_CreateItem("NOINDEX")	Create a IBM Content Manager document.
IBMCM_SetAttributeValue("USERID, @OPERATOR")	Set an attribute value on the IBM Content Manager document.
IBMCM_SetMimeType("application/msword")	Set the MIME type for the IBM Content Manager document you are uploading.
IBMCM_CreateFolder("NOINDEX","123456789")	Create an IBM Content Manager folder that is based on the item type and parent folder ID.
IBMCM_SetFolderAttributeValue("Name","MyFolder")	Set an attribute value on the IBM Content Manager folder.
IBMCM_SetDestinationFolder ("\APT")	Identify the folder into which the uploaded is IBM Content Manager.
IBMCM_UploadDCO_DOC()	Upload the images that are associated with the current Document object of the document hierarchy to IBM Content Manager.

Parent topic: [IBM Content Manager Connector actions](#)

Search and download action attributes of IBM Content Manager

The Datacap Connector for IBM® Content Manager actions can help you to search and download the existing file content.

SearchAndDownload action

Table 1. SearchAndDownload action:

Action	Description
IBMCM_SetSearchAndDownloadCriteria	sets the query criteria: item type, attribute name, comparison operator, attribute value, data type.
IBMCM_SetSearchAndDownloadDirectory	sets the output directory for the downloaded files.
IBMCM_SetSearchAndDownloadMaximum	sets the maximum number of files to download for the batch.
IBMCM_SetSearchAndDownloadSort	determines whether to sort, the sort by attribute, and the sort order (ascending or descending).

Action	Description
IBMCM_SetSearchAndDownloadStatusAttribute	updates a download status attribute for the downloaded item.
IBMCM_SearchAndDownload	uses the parameters set from the actions above to search and download IBM Content Manager files.

Parent topic: [IBM Content Manager Connector actions](#)

Updating IBM Content Manager content with search and download actions

Processes the downloaded files (using the Search and Download action) in Datacap application to find any additional data in the content.

Using the search and download actions, you have downloaded the IBM® Content Manager content onto a directory. You are now ready to process these files in your Datacap application to find any additional data in the content. These new data can then be directly updated as new or updated attributes in the IBM Content Manager repository.

The example in the following table shows the sequence in which you must add the actions for the update scenario.

Table 1. The sequence of actions to update a document's attributes into IBM Content Manager

Action	Description
IBMCM_Logon("ibmcmsrv,userid,password")	logs on to the IBM Content Manager server.
IBMCM_SetSearchAndDownloadCriteria("APT","APT_FLAG","=", "NA", "String")	sets the criteria for the file downloads.
IBMCM_SetSearchAndDownloadDirectory("@APPPATH"(vscanimagedir))	sets the download directory.
IBMCM_SearchAndDownload()	searches and downloads the files.
Vscan.SetSourceDirectory("@APPPATH"(vscanimagedir))	sets the source directory for the scan (must be the same directory from the download).
Vscan.Scan()	scans the files in the directory and sets the PageName.
IBMCM_SetSearchOnlyFolderItems("False")	sets up to search for all items.
IBMCM_SearchItem("", @P.PageName)	searches and sets the current object based on the item ID in PageName.
IBMCM_SetAttributeValue("APT_FLAG", "DONE")	updates the document attributes in IBM Content Manager.
IBMCM_SetAttributeValue("newAttribute", "newAttributeValue")	updates the document attributes in IBM Content Manager.

Parent topic: [IBM Content Manager Connector actions](#)

FileNet P8 Connector actions

The FileNet® P8 Connector actions integrate Datacap applications with IBM® FileNet Content Engine.

You can use FileNet P8 Connector actions to upload documents and index fields into a Content Engine repository.

To use the Secure Socket Layer to encrypt communications between Datacap and the IBM FileNet P8 repository, you must set up an SSL-encrypted connection in the FileNet P8 client.

To set up SSL connection between a .NET client and FileNet P8 server, complete the following steps:-

1. Go to <https://<server>:<port>/wsi/FNCEWS40MTOM/>.
2. Click Continue to this website.
3. Click Certificate error in the address bar.
4. Click View certificate. The Certificate window opens.
5. Click Install certificate, then click OK.
6. On the Certificate Import Wizard window, select Local Machine, and click Next.
7. On the Certificate Store window, select Place all certificates in the following store. Browse to the appropriate location, and click Next.
8. On the Select Certificate Store window, select Trusted People from the list of certificate stores.
9. Verify the settings that you specified, and click Finish.

The following list describes the main functions of the FileNet P8 Connector actions.

- Set up the URL of the Content Engine repository
- Log in to the Content Engine
- Set the class ID of the target location on Content Engine as ObjectStore or FileStore
- Set a locale that is accepted by the IBM FileNet P8 web service
- Set the object ID for the Object Store on Content Engine
- Set the path to the IBM FileNet P8 folder where you are uploading documents
- Specify the content type that defines the fields within a document library for the uploaded documents, such as an Invoice
- Create a folder in Content Engine where you can upload documents
- Specify whether the scanned documents that are exported from Datacap to the FileNet Content Manager inherit security from the parent folder.
- Upload the document, page, or directory into Content Engine
- [FileNet P8 Connector prerequisites](#)
To configure and run FileNet P8 Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 9.0 or later.
- [FileNet P8 Connector settings](#)
Record the system settings that you want to use to configure the FileNet P8 Connector actions and have these values available during the configuration process.
- [Configuring FileNet P8 Connector actions](#)
You must create an Export ruleset and configure its rules and functions with FileNet P8 Connector actions to upload documents from Datacap applications into Content Engine.
- [FileNet P8 Connector upload examples](#)
The Datacap Connector for IBM Content Manager Upload actions configure the connection between the Datacap application and the IBM Content Manager repository.
- [Downloading FileNet P8 content in bulk using a FileNet Sweep Job](#)
If you have a FileNet P8 repository, you can use Datacap to discover additional information that were not extracted as metadata properties.

Parent topic: [Datacap Connector actions](#)

FileNet P8 Connector prerequisites

To configure and run FileNet® P8 Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 9.0 or later.

A complete and up-to-date list of the hardware and software requirements for IBM® Datacap is available on the IBM support site at <http://www.ibm.com/support/docview.wss?uid=swg27043811>.

Supported versions of the following repository clients must be installed on each Datacap computer that runs the export rule set. Export actions are run on Rulerunner in production. Export actions can also be run in Datacap Studio and Datacap Desktop for development or test purposes. Computers that run rules must have the appropriate clients, such as Rulerunner and Datacap Studio, installed on them.

- Datacap Version 9.0 or later installed and running on either a single computer or a client/server installation
- Network access to a supported version of IBM FileNet Content Engine .NET Client

The following repository clients must be installed on the Datacap Web Client that runs the Datacap export process. These clients are run on Microsoft Windows operating systems.

- Microsoft Windows Communications Foundation
- IBM FileNet P8 software package so that you can run the Content Engine Server installation. The Content Engine installation program contains the IBM FileNet Content Engine .NET Client

Parent topic: [FileNet P8 Connector actions](#)

Related information:

[Hardware and Software Requirements for IBM Datacap Version 9.0.1](#)

FileNet P8 Connector settings

Record the system settings that you want to use to configure the FileNet® P8 Connector actions and have these values available during the configuration process.

This table describes the parameters that are required for Datacap Connector for FileNet Content Manager actions.

Table 1. Required IBM FileNet P8 parameter settings

Connector action	Description
Set URL	The URL for the FileNet P8 web service.
Logon	FileNet P8 user ID and password.
Set Target Class ID	The value of the Class ID. Use either ObjectStore or FileStore. The default value is ObjectStore.
Set Locale	The locale value that is accepted by the FileNet P8 web service. Represented by a two-letter language code and a two-letter country code, for example, en_US or de_DE.
Set Target Object ID	The Object ID value to assign to the object store.
Set Security Parentage	Boolean parameter (True or False) that specifies whether the scanned documents that are exported from Datacap to the FileNet Content Manager inherit security from the parent folder. The default value is False, and the security is not inherited from the parent folder.

Connector action	Description
Set Destination Folder	The path to the FileNet P8 folder in the object store where you are uploading the documents, for example \TravelDocs\.
Create Folder	The name of the folder to create for the target class and object.
Set Doc Class ID	The value of the Document Class ID. The default value is Document.
Set Doc Title	The value of a Document Title or a predefined special variable. The default value is Title.
Set Property	The value of the Property ID and the value or predefined special variable to assign to the property.
Set Multiple Page Documents	The parameter that specifies whether the upload actions create a single page or a multiple page document.
Upload Document	None
Upload Page	None
Upload Dir	The full path of the folder that contains the images you want to upload, for example, C:\images,True. Use True to delete the images from the folder after they are uploaded. Use False to leave the images in the folder after they are uploaded.

Parent topic: [FileNet P8 Connector actions](#)

Configuring FileNet P8 Connector actions

You must create an Export ruleset and configure its rules and functions with FileNet® P8 Connector actions to upload documents from Datacap applications into Content Engine.

About this task

Datacap Connector for FileNet Content Manager actions can upload images from a Datacap batch to the IBM® FileNet Content Server library by using the IBM FileNet P8 XML web service.

Procedure

To configure FileNet P8 Connector actions:

1. Install the IBM FileNet P8 Runtime Environment and its prerequisites. For more information, see the IBM FileNet P8 installation instructions.
2. Install the IBM FileNet Content Engine Client files that are provided with the Content Engine Server installation program. The version of the Content Engine Client you install must match the version of the Content Engine Server. Run the installation program that matches the version of the installed Content Engine Server.

Version	Part Number	Installation program
IBM Content Manager 5.1	CI1NIML	5.1.0-P8CE-Win.exe
IBM Content Manager 5.0	CZS02ML	5.0.0-P8CE-Win.exe

3. Verify the URL and the version of the FileNet P8 Server. For example, `http://myp8server:9080/wsi/FNCEWS40MTOM`
4. Add the FileNet P8 Connector actions (FileNetP8.RRX) to the Export rulesets.

Example

The following example describes an Export To P8 ruleset that logs on to the Content Engine server. Then, it uploads a single page document into the Content Engine repository.

The ruleset contains the Connect to CE and AddDocument rules. The Connect to CE rule contains the Logon function and actions you must run to make the connection to Content Engine. The AddDocument rule contains the AddPage function with actions that define the title and format of the page, and upload the page.

Export To P8 ruleset

- Connect to CE rule
 - Logon function
 - FNP8_SetURL("http://MyServer:9080/wsd/FNCEWS40MTOM")
 - FNP8_Login("P8Admin UserID,P8Admin Password")
 - FNP8_SetLocale("en_US")
 - FNP8_SetTargetClassID("ObjectStore")
 - FNP8_SetTargetObjectID("ObjectStoreName")
 - FNP8_SetDestinationFolder("/mydestfolder")
- AddDocument rule
 - AddPage function
 - FNP8_SetDocTitle("@ID")
 - FNP8_SetDocType("TIF")
 - FNP8_Upload()

Parent topic: [FileNet P8 Connector actions](#)

FileNet P8 Connector upload examples

The Datacap Connector for IBM® Content Manager Upload actions configure the connection between the Datacap application and the IBM Content Manager repository.

» You can use these actions to upload the following items and their associated index values from Datacap into Content Engine:

- A single page document
- An image that contains multiple pages

«

The examples in the following tables describe the sequence in which you must add the actions to the Export To P8 ruleset for the upload scenarios.

Upload a single page document

Table 1. The sequence of actions for uploading a single page document into IBM FileNet

Content Engine

Action	»DCO object «	»DCO event «	Description
»Step 1: P8 connection information«			
FNP8_SetURL("http://MyServer:9080/wsd/FNCEWS40MTOM")	Batch	Open	Set the URL for the FileNet® P8 web server.
FNP8_Login("admin,password")	Batch	Open	Provide the Content Engine user login credentials: admin and password.
»Step 2: P8 repository attributes«			
FNP8_SetTargetClassID("ObjectStore")	Docu ment	Open	Set the top-level repository type on Content Engine to ObjectStore.
FNP8_SetTargetObjectID("AP_ObjectStore")	Docu ment	Open	Specify the name of the object store in which to store the document as AP_ObjectStore.
FNP8_SetDestinationFolder("\TravelDocs")	Docu ment	Open	Identify the folder into which the document is uploaded in Content Engine as \TravelDocs.
FNP8_SetLocale("en_US")	Docu ment	Open	Specify en_us as the locale used by the IBM FileNet P8 web service.
»Step 3: Page properties«			
FNP8_SetDocTitle("@ID")	Page	Close	Set the title of the P8 document to be the page's ID, such as TM000001.
FNP8_SetDocType("TIF")	Page	Close	Set the type property for the page to TIF.
»Step 4: Upload the document to P8«			
FNP8_Upload()	Docu ment	Close	Upload the image file for the page to the previously specified destination folder on Content Engine.

Upload a multiple page document

Table 2. The sequence of actions for uploading a multiple page document into IBM FileNet Content Engine

Action	»DCO object «	»DCO event «	Description
»Step 1: P8 connection information«			
FNP8_SetURL("http://MyServer:9080/wsd/FNCEWS40MTOM")	Batch	Open	Establish the URL for the FileNet P8 web service.

Action	»DCO object «	»DCO event «	Description
<code>FNP8_Login("admin,password")</code>	Batch	Open	Provide the Content Engine user login credentials: <code>admin</code> and <code>password</code> .
»Step 2: P8 repository attributes«			
<code>FNP8_SetTargetClassID("ObjectStore")</code>	Docu ment	Open	Set the top-level repository type on Content Engine to <code>ObjectStore</code> .
<code>FNP8_SetTargetObjectID("AP_ObjectStore")</code>	Docu ment	Open	Specify the name of the object store in which to store the document as <code>AP_ObjectStore</code> .
<code>FNP8_SetDestinationFolder("\TravelDocs")</code>	Docu ment	Open	Identify the folder into which the document is uploaded in Content Engine as <code>\TravelDocs</code> .
<code>FNP8_SetLocale("en_US")</code>	Docu ment	Open	Specify the language to use on the IBM FileNet P8 Web Service. For example, enter <code>en_US</code> if you are using US English on the user interface.
»Step 3: Page properties«			
<code>FNP8_SetDocClassID("Document")</code>	Page	Close	Set the FileNet Document Class ID for the page to <code>Document</code> .
<code>FNP8_SetProperty("IndexValue1, \IndexField1")</code>	Page	Close	Set the <code>IndexValue1</code> property value of the P8 document to the value of the field that is named <code>IndexField1</code> on this page.
»Step 4: Upload the document to P8«			
<code>FNP8_Upload()</code>	Docu ment	Close	Upload the multiple page image file for this document to the specified destination folder on Content Engine.

Parent topic: [FileNet P8 Connector actions](#)

Downloading FileNet P8 content in bulk using a FileNet Sweep Job

If you have a FileNet P8 repository, you can use Datacap to discover additional information that were not extracted as metadata properties.

You can use the FileNet sweep framework and its bulk processing capabilities to download the document content into a directory. Once these files are in the directory, a Datacap application can be used to ingest the documents into Datacap and extract information that can be exported back into FileNet P8. See the IBM Knowledge Center for more details about handling bulk processing with FileNet sweeps:

https://www.ibm.com/support/knowledgecenter/en/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/p8pcc175.htm.

This section provides the procedures to create a Sweep Job that downloads the FileNet P8 documents in a format that can be ingested by Datacap. If you decide that a Sweep Policy is more suited for this task, you can modify the JavaScript provided to fit the sweep policy framework as long as you follow the same download file name convention.

File naming convention

You need to update the properties of these documents. Datacap expects the downloaded file names to follow a naming convention that concatenates the document item ID and the original content file name together for each of the downloaded content element.

For Datacap to be able to identify the document that is associated with the downloaded content file, the file name must begin with the document item ID. For example.

{FE52C000-EBF1-4797-BF3B-5CF98AFC5854}.TM000001.tif.

Datacap parses the file name and identify the document ID of the file. This allows the Datacap application to update the existing FileNet P8 document with properties that are found during Datacap processing.

Output directory

In the JavaScript, you also need to specify the download directory. This must be a common directory or shared drive that is readable and writable for both the FileNet P8 and Datacap servers if FileNet P8 and Datacap are not co-located.

- [Creating JavaScript sweep action handler to download document content](#)
- [Creating custom sweep action using the JavaScript](#)
- [Creating custom job sweep referring the action](#)
- [Designing your Datacap application](#)

Parent topic: [FileNet P8 Connector actions](#)

Creating JavaScript sweep action handler to download document content

Adapt the following item in the JavaScript example:

Output directory: Change the output directory in the script. Because the JavaScript code is run on a Content Platform Engine server, the content download directory that you specify must satisfy the following requirements:

- It exists on the server as an existing local directory or a mapped network directory
- It is one to which the server has write permissions.
- It is accessible to the Datacap application that reads the files from this location during the scan task.

JavaScript sweep action script:

```
importPackage(Packages.com.filenet.api.core);
importPackage(Packages.com.filenet.api.constants);
importPackage(Packages.com.filenet.api.exception);
importPackage(Packages.com.filenet.api.sweep);
importPackage(Packages.com.filenet.api.engine);

/* Implement for custom job and queue sweeps. */
```

```

function onSweep(sweepObject, sweepItems) {
    var hcc = HandlerCallContext.getInstance();
    hcc.traceDetail("Entering DatacapSweepHandler.onSweep");
    hcc.traceDetail("sweepObject = " +
        sweepObject.getProperties().getIdValue(PropertyNames.ID) +
        ", sweepItems.length = " + sweepItems.length);

    /* Iterate the sweepItems */
    idx = 0;
    for (idx = 0; idx < sweepItems.length; idx++) {
        /* At the top of your loop, always check to make sure
        * that the server is not shutting down.
        * If it is, clean up and return control to the server.
        */
        if (hcc != null && hcc.isShuttingDown()) {
            throw new EngineRuntimeException(ExceptionCode.E_BACKGROUND_TASK_TERMINATED,
                this.constructor.name + " is terminating prematurely because the server is
                shutting down");
        }

        var item = sweepItems[idx].getTarget();
        hcc.traceDetail("sweepItems[" + idx + "]= " +
            item.getProperties().getIdValue("ID"));

        try {
            var CEObject = Document(item);
            var celist = CEObject.get_ContentElements();
            var docID = CEObject.getProperties().getIdValue("ID");
            for (i = 0; i < celist.size(); i++)
            {
                var ce = celist.get(i);
                var folderName = "X://sweep"; /* output directory */
                this._downloadContent(folderName, docID, ce);
            }

            /* Set outcome to PROCESSED if item processed successfully.*/
            sweepItems[idx].setOutcome(SweepItemOutcome.PROCESSED,
                "item processed by " + this.constructor.name);
        }
        /* Set failure status on objects that fail to process.*/
        catch (ioe) {
            sweepItems[idx].setOutcome(SweepItemOutcome.FAILED, "DatacapSweepHandler: " +
                ioe.rhinoException.getMessage());
        }
    }
    hcc.traceDetail("Exiting DatacapSweepHandler.onSweep");
}

/* private function to download content elements.
* the folderName should use the forward slash "/" as path separator.
* If the file exists it will be overwritten.
*/
function _downloadContent(folderName, docID, ct)
{
    var out = new java.io.FileOutputStream(folderName + "/" + docID + "." +
ct.get_RetrievalName());
    var docLen = ct.get_ContentSize().intValue();
    var buf = java.lang.reflect.Array.newInstance(java.lang.Byte.TYPE, docLen);
    var stream = ct.accessContentStream();

    stream.read(buf, 0, docLen);
    out.write(buf);
    out.flush();
    stream.close();
}

```

```

        out.close();
    }

    /* Called automatically when the handler is invoked by a custom sweep job
    * or sweep policy. Specify properties required by the handler, if any.
    * If you return an empty array, then all properties are fetched.
    */
    function getRequiredProperties() {
        var pnames = [];
        return pnames.toString();
    }

    /* Implement for custom sweep policies.
    * This method is not implemented because this is an example of a custom sweep job.
    */
    function onPolicySweep (sweepObject, policyObject, sweepItems)
    {}

```

[Go to next step](#)

Parent topic: [Downloading FileNet P8 content in bulk using a FileNet Sweep Job](#)

Creating custom sweep action using the JavaScript

About this task

This task is for creating custom sweep action using the JavaScript.

Procedure

To create a custom sweep action, do the following steps:

1. Start the administration console (FileNet P8 ACCE).
2. In the domain navigation pane, click the object store.
3. In the object store navigation pane, open the Sweep Management > Sweep Actions folder.
4. Click the New button in the Sweep Action tab.
5. Enter a name for the custom sweep action and click Next.
6. Select Action type JavaScript, and click Next.
7. Enter the JavaScript code from the previous section into the Script text. Modify the *folderName* variable to your download directory. Use the double forward slash as the path separator, for example C://sweep//input. Click Next.
8. Finish the wizard. You have created a custom sweep action.

[Go to next step](#)

Parent topic: [Downloading FileNet P8 content in bulk using a FileNet Sweep Job](#)

Creating custom job sweep referring the action

About this task

With the sweep action that is defined in the previous section, now you are ready to create a sweep job to use the custom sweep action. You also specify the sweep targets or criteria that identify the documents that the custom sweep action applies to. For the Datacap scenario, the criteria should identify all the documents that are processed by the Datacap application.

Procedure

To create custom sweep action follow the following steps:

1. Start the administration console (FileNet P8 ACCE).
2. In the domain navigation pane, click the object store.
3. In the object store navigation pane, open the Sweep Management > Job Sweeps > Custom Jobs folder.
4. Click New in the Custom Jobs tab.
5. Enter a name for the sweep job, change the Sweep mode to Preview for testing the query script, or Normal to run. Do not enable custom sweep yet, and click Next.
6. Define the sweep target class, example that is given, Document. Define the filter expression, example given *IsCurrentVersion=True AND Document title LIKE 'Invoice%'*. [Optional] Enable Include subclasses and Record failures. Select the sweep action that we defined in the previous section from the choice list drop down, and click Next
7. Do not define any start or stop dates, so that the job can be run immediately. Click Next.
8. Verify the summary and click Finish
9. Wait for the custom job to be created.
10. To test the Job sweep, open the custom job, click Enable for status, [Optional] go to the Properties tab and change the batch limit, Click Save, and Refresh
11. Scroll down to view the Examined, Processed, and Failed object counts.
12. Check the results or errors in the Sweep Results tab.
13. If you had chosen the Preview sweep mode, and there were no errors in the Sweep Results tab, you can clone the sweep job and change the sweep mode to 'Normal' to run the actual job sweep.
14. If you had chosen the Normal sweep mode, look at the output directory that you specified in the JavaScript to verify that the document content was downloaded as expected.
15. You are ready to run your Datacap application.

[Go to next step](#)

Parent topic: [Downloading FileNet P8 content in bulk using a FileNet Sweep Job](#)

Designing your Datacap application

With the FileNet P8 document content files that are downloaded to a directory from the FileNet sweep job, you are ready to import them into Datacap for processing.

Code snippet of a sample Datacap application that ingests files from the output directory and updates the existing FileNet P8 document with updated property values. See the FileNet P8 actions topic [FileNet P8 actions](#) for more details.

Prerequisite

The downloaded image files from FileNet P8 with file name is in the format of itemID.tif, For example, {FE52C000-EBF1-4797-BF3B-5CF98AFC5854}.TM000001.tif.

Sample FileNet P8 actions to update the FileNet P8 document properties:

```
[Batch level]
VScan.SetSourceDirectory("X://sweep")
VScan.Scan()
```

```
[Page level-open]
FileNetP8.FNP8_SetKeyProperty("ID,@P.PageName")
```

```
[Field level]
FileNetP8.FNP8_SetProperty("myNewPropertyName, @myNewPropertyValue ")
```

```
FileNetP8.FNP8_SetProperty("myNewPropertyName2, @myNewPropertyValue2")  
...
```

```
[Page level-close]  
FileNetP8.FNP8_UpdateProperties()
```

@P.PageName variable

The Scan() action populates the page variable @P.PageName, to be the file name of the image file without the file extension. For example, the image file {FE52C000-EBF1-4797-BF3B-5CF98AFC5854}.TM000001.tif has a PageName of "{FE52C000-EBF1-4797-BF3B-5CF98AFC5854}". As you can see, this page variable is used later in the FNP8_SetKeyProperty() to identify the FileNet item ID.

Parent topic: [Downloading FileNet P8 content in bulk using a FileNet Sweep Job](#)

Documentum Connector actions

The Documentum Connector actions integrate Datacap applications with a Documentum Docbase content repository.

The Documentum Connector actions integrate Datacap applications with the Documentum Docbase content repository. You can then use the Documentum Connector actions to upload documents and index fields into a Documentum repository.

The following list describes the main functions of the Documentum Connector actions.

- Log in to the Documentum repository
- Specify the content type or format in which to release documents to the Documentum repository, such as TIF or PDF
- Set the name of the folder in Datacap from which to upload the documents into the Documentum repository
- Set the object name for the file that is uploaded into the Documentum repository
- Upload the indexed documents or pages into the Documentum repository
- [Documentum Connector prerequisites](#)
To configure and run Documentum Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 8.0, 8.0.1, and 9.0.
- [Documentum Connector settings](#)
Record the system settings that you want to use to configure the Documentum Connector actions and have these values available during the configuration process.
- [Configuring Documentum Connector actions](#)
You must create an Export ruleset and configure its rules and functions with Documentum Connector actions to upload documents from Datacap applications into Content Engine.
- [Documentum Connector upload examples](#)
The Documentum Connector Upload actions configure the connection between the Datacap application and the Documentum Docbase repository.

Parent topic: [Datacap Connector actions](#)

Documentum Connector prerequisites

To configure and run Documentum Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 8.0, 8.0.1, and 9.0.

The following repository clients must be installed on each Datacap computer that runs the export ruleset. Export actions are run on Rulerunner in production. Export actions can also be run in Datacap Studio or Datacap Desktop for development or test purposes. Computers that run rules must have the appropriate clients, such as Rulerunner and Datacap Studio, installed on them.

- Datacap Version 8.0, 8.0.1, 9.0, or 9.0 installed and running on either a single computer or a client/server installation
- Network access to Documentum Foundation Classes 6.6, DFC Version 2.2.5.219 SP2 or later
- Network access to Documentum DFC Runtime Environment: not installed with Datacap. See the Documentum documentation for information about the installation, operating system requirements, and configuration. You must restart the server after you install this component.

Parent topic: [Documentum Connector actions](#)

Related information:

[Hardware and Software Requirements for IBM Datacap Version 9.0.1](#)

Documentum Connector settings

Record the system settings that you want to use to configure the Documentum Connector actions and have these values available during the configuration process.

This table describes the parameters that are required for Documentum Connector actions.

Table 1. Required Documentum Docbase parameter settings

Connector action	Description
Logon	Domain name, server name, user ID, password
Set Content Type	The content type that is defined in the repository for the object, for example TIFF, JPEG, DOC.
Set Folder Name	The name of the Datacap folder from which the file is uploaded.
Set Object Name	The Object Name value for the file that you are uploading to the repository.
Upload Document	Upload all of the pages in the document.
Upload Page	Upload the selected page from the document.

Parent topic: [Documentum Connector actions](#)

Configuring Documentum Connector actions

You must create an Export ruleset and configure its rules and functions with Documentum Connector actions to upload documents from Datacap applications into Content Engine.

About this task

To upload Datacap scanned images to a Documentum repository, configure the Documentum Connector actions on computers that are clients to the Documentum repository. If you have multiple computers that run the Documentum Connector actions, the actions must be configured the same way on each computer.

Procedure

To configure Documentum Connector actions:

1. Install the Documentum DFC Runtime Environment on the computers that are going to run Documentum Connector actions. See the Documentum DFC Runtime installation documentation for instructions.
2. Restart Datacap after the Documentum DFC Runtime installation.
3. Add Connector actions to the ExportDocumentumByPage and ExportDocumentumByDocument rulesets.

Example

The following example describes an ExportDocumentumByPage ruleset that logs on to the Documentum server and uploads a single page file into Documentum repository.

The ruleset contains the Login and Page Upload rules, functions, and actions. The Login rule is bound to the batch level. The PageUpload rule is bound to the page level.

The Login function logs in the application to Documentum. The PageUpload function contains actions that set the format of the page to TIFF and define the folder from which the page is uploaded. These actions also set the object name to use for the page and upload the page to Documentum Docbase.

ExportDocumentumByPage ruleset

- Login rule
 - Login function
 - DM_Logon("dmsrv","userid","password")
- Page Upload rule
 - Page Upload function
 - DM_SetContentType("tiff")
 - DM_SetFolderName("/MyDocument")
 - DM_SetObjectName("@ID")
 - DM_UploadPage()

Parent topic: [Documentum Connector actions](#)

Documentum Connector upload examples

The Documentum Connector Upload actions configure the connection between the Datacap application and the Documentum Docbase repository.

Use the Documentum Connector actions to upload a single page file into Documentum Docbase. You can also upload an image that contains multiple pages and their associated index values from Datacap.

The examples in the following table describe the sequence in which you must add the actions to the ruleset for this upload scenario.

Upload a single page file

Table 1. The sequence of actions for uploading a single page document file into Documentum.

Action	Description
DM_Logon ("Domain", "Servername", "Userid", "Password")	Log the Datacap Client on to the Documentum Content Server that links Datacap to the repository that you specify as a parameter. A rule with the DM_Logon action must begin your Documentum upload procedures.
DM_SetContentType ("tiff")	Assigns the Content Type to be TIFF for the page that you are uploading to the repository

Action	Description
DM_SetFolderName ("/MyPage")	Name of the target folder on Documentum to which the file is uploaded. This parameter can be the path to the folder with forward slashes or the ObjectID of the folder as defined in Documentum. To get the ObjectID of the folder, you can use the Documentum client to view the folders on Documentum.
DM_SetObjectName ("@ID")	Sets the Object Name value to the uploaded file. If you use this action when you upload the file, the file name is changed to this name in Documentum. For example, if the file name was XYZ when you uploaded the page, the page is named XYZ in Documentum.
DM_UploadPage ()	Uploads the images that are associated with the current Page object of the document hierarchy to Documentum repository.

Upload a multiple page file

You can upload a single document that consists of a multiple TIFF pages by using the DM_UploadDocument action on the document level.

If you want to merge the pages of the document into multiple TIFF files, you can use a rule that contains TifMerge actions to combine multiple pages image files into a file. Then, you can assign that multiple image file to one document. For more information about TifMerge actions, see the TifMerge reference topic.

You can use a subsequent rule that uploads the multiple page document file to a specified Documentum repository.

You can create a single RuleSet Type with actions from both actions files, or you can create rules from different RuleSet Types. In either case, the actions must be similar to the following sequence.

Table 2. The sequence of actions for uploading a multiple page document file into Documentum.

Action	Description
TifMerge_SetFileName ("@ID", ".tif")	Optional: Define the name of the multiple TIFF files that is uploaded to the Documentum repository.
TifMerge_MergeImages ("all")	Optional: During processing, merge all the images of the pages that are associated with the current document.
DM_Logon ("Domain", "Servername", "Userid", "Password")	Log the Datacap Client on to the Documentum Content Server that links Datacap to the repository that you specify as a parameter. A rule with the DM_Logon action must begin your Documentum upload procedures.
DM_SetContentType ("tiff")	Assign the Content Type to be TIFF for the page that you are uploading to the repository.
DM_SetFolderName ("/MyPage")	Name of the target folder on Documentum to which the file is uploaded. This parameter can be the path to the folder with forward slashes or the ObjectID of the folder as defined in Documentum. To get the ObjectID of the folder, you can use the Documentum client to view the folders on Documentum.
DM_SetObjectName ("@ID")	Specifies the Object Name value name of the final uploaded file. This example assigns the value in a smart parameter.
DM_UploadDocument ()	Upload the document and its multiple page file to the Documentum repository.

Parent topic: [Documentum Connector actions](#)

Related information:

[TifMerge](#)

SharePoint Connector actions

The Datacap Connector for Microsoft SharePoint actions integrate Datacap applications with Microsoft Office SharePoint Services (MOSS) for Microsoft SharePoint 2007 and 2010.

You then use SharePoint Connector actions to upload documents and index fields into a SharePoint library.

The following list describes the main functions of the SharePoint Connector actions.

- Log in to the SharePoint library
- Identify and set up the URL of the SharePoint library
- Specify the content type that defines the fields within a document library for the uploaded documents, such as an Invoice
- Set the format in which to release documents to the SharePoint library, such as TIF or PDF
- Create a folder in the SharePoint into which you upload documents
- Set the column properties (index values) in SharePoint for the documents you want to upload
- Upload the indexed documents into the SharePoint library
- [SharePoint Connector prerequisites](#)
To configure and run SharePoint Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 8.0, 8.0.1, and 9.0.
- [SharePoint Connector settings](#)
Record the system settings that you want to use to configure the SharePoint Connector actions and have these values available during the configuration process.
- [SharePoint and Datacap](#)
You create SharePoint columns at the library level, not at the folder level. Datacap passes Datacap index values to these columns.
- [Configuring SharePoint Connector actions](#)
Create an Export ruleset and configure its rules and functions with SharePoint Connector actions. Then, you can upload documents from Datacap applications into a SharePoint library.
- [SharePoint Connector upload examples](#)
The SharePoint Connector Upload actions configure the connection between the Datacap application and the SharePoint library.

Parent topic: [Datacap Connector actions](#)

SharePoint Connector prerequisites

To configure and run SharePoint Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 8.0, 8.0.1, and 9.0.

The repository clients must be installed on each Datacap computer that runs the export ruleset. Export actions are run on Rulerunner in production. Export actions can also be run in Datacap Studio or Datacap Desktop for development or test purposes. Computers that run rules must have the appropriate clients, such as Rulerunner and Datacap Studio, installed on them.

You must meet the following prerequisites to export images to a SharePoint library:

- Datacap Version 8.0, 8.0.1, 9.0, or 9.0 installed and running on either a single computer or a client/server installation

- Network access to a SharePoint 2010 Server or a SharePoint 2007 Server with Microsoft Office SharePoint Services (MOSS) 3.0 installed
- SharePoint URL or HTTP address of every library to which you are releasing images
- Details about the columns in the library where you are exporting images; such as the static names of the columns, the column types, and column restraints
- Valid Content Types for each library and the exact spelling of these Content Types.
- The user ID that logs in to SharePoint is different than the user ID on the computer where the Export task runs. Then, you need the SharePoint login credentials.

Parent topic: [SharePoint Connector actions](#)

Related information:

[Hardware and Software Requirements for IBM Datacap Version 9.0.1](#)

SharePoint Connector settings

Record the system settings that you want to use to configure the SharePoint Connector actions and have these values available during the configuration process.

This table describes the parameters that are required for Datacap Connector for Microsoft SharePoint actions.

Table 1. Required SharePoint Connector parameter settings

Connector action	Description
Create Folder	The folder in the SharePoint library into which you import your documents.
Set URL	The URL address of the SharePoint library.
Login	User ID, password, optional SharePoint domain.
Set Content Type	The name of the Content Type that defines the fields within a document library for the uploaded documents, such as an Invoice
Set File Type	The format in which to upload the document to the SharePoint library, for example TIF or PDF.
Set Property	The column property in SharePoint for the documents you want to upload.
Upload Batch	None
Upload Document	None
Upload Page	None
Upload Dir	The full path of the folder that contains the images you want to upload, for example, C:\images,True. Use True to delete the images from the folder after they are uploaded. Use False to leave the images in the folder after they are uploaded.

Parent topic: [SharePoint Connector actions](#)

SharePoint and Datacap

You create SharePoint columns at the library level, not at the folder level. Datacap passes Datacap index values to these columns.

The following table identifies the relationships between SharePoint columns and the index values passed to SharePoint columns by Datacap.

SharePoint Column Type	Datacap Index Value
Column constraints	<p>Construct your Datacap application so it produces and exports index values that are valid according to the column constraints of SharePoint.</p> <p>If the index field value passed to SharePoint does not fit within the SharePoint constraints, the upload to SharePoint fails. SharePoint messages are logged in the SPEXport_rrs.log file in the batch folder.</p>
When a column is defined as required in SharePoint	<p>Each required column in a SharePoint library must be set up with a default value defined.</p> <p>Ensure that the index value for a required SharePoint column is always exported by the Datacap application.</p> <p>If during Datacap processing the Operator overrides a required field and an empty field is passed to SharePoint. The upload to SharePoint fails if no default value is defined for the column. SharePoint messages are logged in the SPEXport_rrs.log file in the batch folder.</p>
Single line of text columns (index fields)	The index value must be a text string. The index value for this type of column can contain special characters, for example, !@#\$%^&*()_< >.
Might or might not have the Maximum number of characters	Ensure that the exported index value does not contain more than the maximum number of characters that are allowed for SharePoint columns.
Multiple lines of text	Same as single line of text.
Choice (multi-value list)	Define the default value to use when Allow Fill-In Choices is set to No.
Number (integer, float)	
Currency	
Yes/No columns	<p>The exported index value must be one of the following values:</p> <ul style="list-style-type: none"> • 0 for No • 1 for Yes
Date or Data and Time	<p>The exported index value must be one of the following values:</p> <ul style="list-style-type: none"> • YYYY-MM-DD • YYYY-MM-DDTHH:MM:SSZ (T and Z must enclose the time stamp)
Lookup	Not supported.
Calculated	Not supported
Business data	Not supported
Hyperlink or picture	Ensure that the exported index value is a valid URL address

SharePoint Column Type	Datacap Index Value
Document Library Versioning Settings - Document Version History must be set to No versioning	Does not support version history.
Person or Group	Cannot be exported, automatically assigned by SharePoint.

Parent topic: [SharePoint Connector actions](#)

Configuring SharePoint Connector actions

Create an Export ruleset and configure its rules and functions with SharePoint Connector actions. Then, you can upload documents from Datacap applications into a SharePoint library.

About this task

You can export documents from an Datacap batch to a SharePoint library by adding the SharePoint Connector actions to the Export rulesets.

Procedure

To configure SharePoint Connector actions:

1. Verify the URL of the SharePoint library.
2. Add the SharePoint connector actions (SPExport.RRX) to the Export rulesets.

Example

The following example describes an Export To SP ruleset that logs on to SharePoint and uploads a single page document into the SharePoint library.

The ruleset contains the Connect to SP and AddDocument rules. The Connect to SP rule contains the Logon function and actions you must run to make the connection to the SharePoint library. The AddDocument rule contains the AddPage function with actions that define the title and format of the page, and upload the page.

Export To SP ruleset

- Connect to SP rule
 - Logon function
 - SP_Login("userID,password,domain")
 - SP_SetURL("http://blue/Docs/Documents/+BatchID+/@ID")
 - SP_CreateFolder("http://blue/Docs/Documents/Test")
 - SP_Property("Date,@Value")
- AddDocument rule
 - AddPage function
 - SP_SetContentType("Invoice")
 - SP_SetFileType("jpg")
 - SP_Upload()

Parent topic: [SharePoint Connector actions](#)

SharePoint Connector upload examples

The SharePoint Connector Upload actions configure the connection between the Datacap application and the SharePoint library.

The Datacap Connector for Microsoft SharePoint Upload actions configure the connection between the Datacap application and the SharePoint library. You use these actions to upload a single page file or an image that contains multiple pages and their associated index values from Datacap into SharePoint.

The examples in the following tables describe the sequence in which you must add the actions to the Export To SharePoint ruleset for the upload scenarios.

Upload a single scanned image file

Table 1. The sequence of actions for uploading a single scanned image file into SharePoint.

Action	Description
<code>SP_SetURL("http://full.url.com")</code>	Establish the URL for the SharePoint library.
<code>SP_Login("admin,password")</code>	Provide the SharePoint user login credentials: admin and password.
<code>SP_Upload()</code>	Upload the image file for the page to the specified URL location on the SharePoint library.

Upload a batch of scanned images

Table 2. The sequence of actions for uploading a batch of scanned images into a SharePoint library.

Action	Description
<code>SP_SetURL("http://full.url.com")</code>	Establish the URL for the SharePoint library.
<code>TifMerge_SetFileName("@ID",".tif")</code>	Optional: Define the name of the multiple TIFF files that is uploaded to the Documentum Docbase.
<code>TifMerge_MergeImages("all")</code>	Optional: During processing, merge all the images of the pages that are associated with the current document.
<code>SP_SetContentType("tiff")</code>	Assign the Content Type to be TIFF for the page that you are uploading to the repository.
<code>SP_Login("admin,password")</code>	Provide the SharePoint user login credentials: admin and password.
<code>SP_Upload()</code>	Upload the image file for this batch to the specified URL location on the SharePoint library.

Upload pre-scanned images

Table 3. The sequence of actions for uploading pre-scanned images into a SharePoint library.

Action	Description
<code>SP_SetURL("http://full.url.com")</code>	Establish the URL for the SharePoint library.

Action	Description
SP_SetProperty("Date,@Value")	Set an index value for the Date column in SharePoint.
SP_Login("admin,password")	Optional: Provide the SharePoint user login credentials: admin and password.
SP_UploadDIR("/MyImages")	Upload the image files in this directory to the specified URL location on the SharePoint library. Specify whether the files are uploaded or deleted.

Collect field data and populate SharePoint columns

Table 4. The sequence of actions for collecting field data and populating SharePoint columns.

Action	Description
SP_SetURL("http://full.url.com")	Establish the URL for the SharePoint library.
TifMerge_SetFileName("@ID", ".tif")	Optional: Define the name of the multiple TIFF files that is uploaded to the Documentum Docbase.
TifMerge_MergeImages("all")	Optional: During processing, merge all the images of the pages that are associated with the current document.
SP_SetProperty("Date,@Value")	Set an index value for the Date column in SharePoint.
SP_SetContentType("tiff")	Assign the Content Type to be TIFF for the page that you are uploading to the repository.
SP_Login("admin,password")	Provide the SharePoint user login credentials: admin and password.
SP_Upload()	Upload the image file for this batch to the specified URL location on the SharePoint library.

Parent topic: [SharePoint Connector actions](#)

FileNet Image Services Connector Connecting actions

You use Datacap Connector for FileNet® Image Services actions to upload documents and commit images to an IBM® FileNet Image Services library.

The Rulerunner Service task that applies FileNet Image Services Connector rules and remembers the images that were previously committed to FileNet. When the Rulerunner Service task runs FileNet Image Services Connector Upload procedures, these previously uploaded images are not recommitted. The Rulerunner Service task generates a separate and unique Page file (*<upload>.xml*) every time it uploads a FileNet document. When the actions to upload documents and commit images to an IBM FileNet Image Services components initialize, the task polls the active batch folder for this Page file. If it does not find the file, it creates a new Page file.

The main function of the FileNet Image Services Connector actions to upload documents and commit images to an IBM FileNet Image Services library:

- Access and open an IBM FileNet Image Services library
- Create a FileNet document to upload into the library
- Define an Index Map that links FileNet properties to values that are associated with objects of the Document Hierarchy
- Associate images with FileNet documents

- Upload indexed documents and images for commitment to the library
- [FileNet Image Services Connector prerequisites](#)
To you configure and run FileNet Image Services Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 8.0, 8.0.1, and 9.0.
- [FileNet Image Services Connector settings](#)
Record the system settings that you want to use to configure the FileNet Image Services Connector actions and have these values available during the configuration process.
- [Configuring FileNet Image Services Connector actions](#)
You must create an Export ruleset and configure its rules and functions with FileNet Image Services Connector actions to upload documents from Datacap applications into a FileNet Image Services library.
- [FileNet Image Services Connector upload examples](#)
The FileNet Image Services Connector Upload actions configure the connection between the Datacap application and the FileNet Image Services library.

Parent topic: [Datacap Connector actions](#)

FileNet Image Services Connector prerequisites

To you configure and run FileNet® Image Services Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 8.0, 8.0.1, and 9.0.

The following repository clients must be installed on each Datacap computer that runs the export ruleset. Export actions are run on Rulerunner in production. Export actions can also be run in Datacap Studio or Datacap Desktop for development or test purposes. Computers that run rules must have the appropriate clients, such as Rulerunner and Datacap Studio, installed on them.

- Datacap Version 8.0, 8.0.1, 9.0, or 9.0 installed and running on either a single computer or a client/server installation
- IBM® FileNet IDM Desktop client
- Network access to the IBM FileNet Image Services library

Parent topic: [FileNet Image Services Connector Connecting actions](#)

Related information:

[Hardware and Software Requirements for IBM Datacap Version 9.0.1](#)

FileNet Image Services Connector settings

Record the system settings that you want to use to configure the FileNet® Image Services Connector actions and have these values available during the configuration process.

This table describes the parameters that are required for Datacap Connector for FileNet Image Services actions.

Table 1. Required FileNet Image Services Connector parameter settings

Action	Description
Library Initialize (IS)	Elements of a previously defined library name by using the syntax <i><DefaultIMS>:<domain>:<organization></i> For example, ISLibrary:Datacap:FileNet
Library Logon	User ID, password
FileNet Database ADO Connect	None

Action	Description
New Document	The name of a previously defined Document Class
Add All Images to Document	None
Create Folder	The name of the new folder, for example Taxes2011
Get Top Folders	None
Save Doc To Folder	The folder name that is preceded by a forward slash, for example /Taxes2011
Add TIF Image To Folder	None
Add PDF Image To Folder	None
Add File To Document	The path name and file name of the file to add to the document, for example C:\Datacap\MQSW\Process\FNLog.log
FileNet Doc ID Set Value	The name of the child field object to which you want to assign the FileNet Document ID.
Use Indexes ON	None
Use Indexes OFF	None
Index Property ID Date Component	<p>The following 4 values:</p> <ol style="list-style-type: none"> 1. The name of the Date property 2. Name of a Document Hierarchy object with a Date property 3. Format of the Date when supplied to the FileNet document 4. Format value of the Date value that is added to the processing index of the task <p>For example,</p> <pre>IndexProperty_ID_Date_Component (FNStart,1040EZ,mmddy,yyyymmdd)</pre>
Index Property Left Justify	<p>Name of the FileNet document property to left align and the maximum size of the value.</p> <p>For example, FNfldData, 256</p>
Index Property Right Justify	<p>Name of the FileNet document property to right align and the maximum size of the value.</p> <p>For example, FNfldData, 256</p>
Upload	None

Parent topic: [FileNet Image Services Connector Connecting actions](#)

Configuring FileNet Image Services Connector actions

You must create an Export ruleset and configure its rules and functions with FileNet® Image Services Connector actions to upload documents from Datacap applications into a FileNet Image Services library.

About this task

You can export documents from an Datacap batch to an IBM® FileNet Image Services library by adding the FileNet Image Services Connector actions to the Export rulesets.

Procedure

To configure FileNet Image Services Connector actions:

1. Verify the URL of the library into which you want to export documents.
2. Add the FileNet Image Services Connector actions (FileNetIDM.RRX) to the Export rulesets.

Example

The following example describes an Export To IS ruleset that logs on to FileNet Image Services and uploads a single page document into the library.

The ruleset contains the Connect to IS, CreateDocument, and AddDocument rules. The Connect to IS rule contains the Logon function and actions you must run to make the connection to the FileNet Image Services library. The AddDocument rule contains the AddPage function with actions that define the title and format of the page, and upload the page.

Export To IS ruleset

- Connect to IS rule
 - Logon function
 - Library_IS_Initialize(ISLibrary:Datacap:FileNet)
 - Library_Login("userID,password")
- AddDocument rule
 - AddPage function
 - NewDocument("1040EZtwo")
 - AddFileToDocument(C:\Datacap\MSQW\Process\FNLog.log)
 - Upload()

Parent topic: [FileNet Image Services Connector Connecting actions](#)

FileNet Image Services Connector upload examples

The FileNet® Image Services Connector Upload actions configure the connection between the Datacap application and the FileNet Image Services library.

Upload a single page file

Use these actions to upload a single page file or a document that contains multiple pages from Datacap into the FileNet Image Services library.

The examples in the following tables show the sequence in which you must add the FileNet Image Services Connector actions to the Export ruleset for the upload scenarios.

Table 1. The sequence of actions for uploading a single page file into an FileNet Image Services library.

Action	Description
Library_IS_Initialize (ISLibrary:Datacap:FileNet)	Initialize the previously defined FileNet Image Services library.

Action	Description
Library_Login("userid, password")	Log in to FileNet Image Services library.
FileNetDB_ADOConnect()	Establish an Active X Data Connection object (ADO) with the specified FileNet database.
NewDocument(1040EZtwo)	Set up a new FileNet document and specify the FileNet Document Class to assign to the new document.
CreateFolder(IncomeTaxes_2011)	Create a top-level FileNet folder in the FileNet Image Services library.
Upload()	Import the document into the FileNet Image Services library.
SaveDocToFolder(IncomeTaxes_2011)	Put the document in the specified folder in the FileNet Image Services library.

Upload a multiple page file

Table 2. The sequence of actions for uploading a multiple page file into an FileNet Image Services library.

Action	Description
Library_IS_Initialize (ISLibrary:Datacap:FileNet)	Initialize the previously defined FileNet Image Services library.
Library_Login("userid, password")	Log in to FileNet Image Services library.
FileNetDB_ADOConnect()	Establish an Active X Data Connection object (ADO) with the specified FileNet database.
NewDocument(1040EZtwo)	Set up a new FileNet document and specify the FileNet Document Class to assign to the new document.
AddAllImagesToDocument()	Assigns all of the images that are associated within the Document object of the Document Hierarchy to the new document.
CreateFolder(IncomeTaxes_2011)	Create a top-level FileNet folder in the FileNet Image Services library.
Upload()	Import the document into the FileNet Image Services library.
SaveDocToFolder(IncomeTaxes_2011)	Put the document in the specified folder in the FileNet Image Services library.

Parent topic: [FileNet Image Services Connector Connecting actions](#)

Email Connector actions

Email Connector actions create Datacap batches from the documents that you receive as email attachments. You can also send email notification messages when specific events occur.

The Email Connector actions option contains the following actions libraries:

- IMAP Email Input Actions (IMail.RRX)
- Exchange Web Service Email Input Actions (EWSMail.RRX)
- Email Sending Actions (EMail.DLL)

- [Email Input actions](#)
Email Input actions scan an email inbox for incoming mail messages and place selected messages and attachments into a new Datacap batch for processing.
- [Email Send actions](#)
You configure the Email Send actions for a Datacap task to compose and send informational email messages under the conditions that you define.
- [Email Connector prerequisites](#)
To configure and run Email Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 8.0, 8.0.1, and 9.0.
- [Email Connector settings](#)
Record the system settings that you want to use to configure the Email Connector actions and have these values available during the configuration process.
- [Configuring Email Connector actions](#)
Email Connector can actions to scan Email servers that support IMAP protocol or Exchange Web Service (EWS) for incoming email messages with attachments. You specify and export the attachments into a batch.
- [Email Connector import examples](#)
The Email Connector actions connect to the Email servers that support IMAP protocol or Exchange Web Service (EWS). These actions scan incoming email messages for attachments that contain the types of attachments that you specify and import the attachments into a batch.

Parent topic: [Datacap Connector actions](#)

Email Input actions

Email Input actions scan an email inbox for incoming mail messages and place selected messages and attachments into a new Datacap batch for processing.

Datacap supports two methods of accessing a mail server to obtain the image attachments.

- The actions in IMail.RRX use the Internet Message Access Protocol (IMAP) for Microsoft Exchange Server, Novell GroupWise, and other mail servers that provide support for IMAP.
- The actions in EWSMail.RRX use Exchange Web Services (EWS) for Microsoft Exchange Server. This service is a SOAP-based method of communication.

The Input actions are typically assigned to a Datacap task that is run by an unattended Rulerunner Service station. These Input actions scan one or more inboxes and defines the type of attachments to include in the batch. For example, you can specify only TIFF images or only PDF files, which helps eliminate input of improper files.

The batch is created when the maximum number of documents is received from the email servers or a specified time interval elapsed. At the batch level, the EmailCount is captured. Each document in the batch is associated with one email message. The document contains a page for each attachment file and variables with the email Subject, From, To, DateSent, Priority, and Body. In addition, the IMail actions capture User, and the EWSMail actions capture DateReceived.

When an email is successfully processed and the attachment is input into the Datacap batch. The email is moved from the inbox to a specified email done folder. If the attachment is not one of the expected types such as a TIFF file, or there is a processing problem. The email, and attachment does not become part of the batch and are moved to a specified email problem folder.

Attention: To further process of the body of an email, or an attachment other than TIF after it was added to a Datacap batch by using the Email actions. You might have to construct more rules and to license other Datacap options.

To capture MS Office document attachments and turn them into images that can be processed by using the Datacap standard Recognition or Verify tasks. You can license the eDocument Conversion actions and put them in the processing rules of your application.

If you capture non-image files by using the Email Input actions, the pages that are created have non-image files that are attached to them. If you capture only the email body, the page might not have any files that are attached. In either of these cases, your application must handle these documents and pages. Processing a batch that contains pages with no images attached by using the standard recognition actions or the verify task, might not work as you expect.

Parent topic: [Email Connector actions](#)

Email Send actions

You configure the Email Send actions for a Datacap task to compose and send informational email messages under the conditions that you define.

You can send notification emails directly to multiple recipients or use the CC and BCC options. The subject line can be specified and the email can be sent with or without attachments. The Send Action is useful when the other actions encounter exceptions and must notify a user. For example, if an error occurs during data verification, you can alert an administrator to take the appropriate action. The items that caused the failure can be attached to the email. If export rules encounter a problem and a batch is not exported successfully, an email can be composed and sent that contains the details.

Parent topic: [Email Connector actions](#)

Email Connector prerequisites

To configure and run Email Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 8.0, 8.0.1, and 9.0.

The following components must be installed and running on your system before you can use Email Connector actions:

- Datacap Version 8.0, 8.0.1, 9.0, or 9.0 installed and running on either a single computer or a client/server installation

For a list of the hardware and software requirements, see <http://www-01.ibm.com/support/docview.wss?uid=swg27020397>.

Email Input actions prerequisites

To use the Datacap Email Input actions, you must have access to a mail server with one of the following servers:

- Exchange Web Service (EWS) enabled on Microsoft Exchange. Datacap recommends that SSL is configured. Use the input actions in EWSMail.RRX.
- Email server that supports IMAP protocol and the mail server and firewall (if any) have IMAP access that is enabled. SSL is not supported now. Use the input actions in IMail.RRX.

You must have an email account that you can use for which you know the server URL, and login user name and password.

The dedicated email account contains an Inbox folder, a *Done* folder for messages that are successfully imported, and a *Problem* folder for messages that encountered errors. The names that you assign to these folders can be specified by using the `im_done_folder` and `im_problem_folder`, or `ex_done_folder` and `ex_problem_folder` actions.

Optional: set up an inbox that is dedicated to the emails that you intend to process. If there are preexisting messages in the Inbox, the oldest messages are processed first.

Email Send actions prerequisites

To use the Datacap Email Sending actions in IMail.RRX ensure:

- You have access to an SMTP server that can relay the emails that are created by the Datacap application.
- The Email Sending actions are run on a computer on which the Windows CDOSYS object or Microsoft Outlook object are registered. The Email Sending actions use the CDOSYS object by default to send email. If the CDOSYS object is not available, the actions can use the Outlook object. The CDOSYS object is available under Windows XP, Windows 2000, Windows 2003 Server or later.
- The Email Sending actions are run on a Workstation and under a user account that has permission to relay or send emails.

Parent topic: [Email Connector actions](#)

Related information:

[Hardware and Software Requirements for IBM Datacap Version 9.0.1](#)

Email Connector settings

Record the system settings that you want to use to configure the Email Connector actions and have these values available during the configuration process.

Each email message that contains one or more of the wanted attachment types becomes a new Datacap document. Message headers and body, and pages are created for each of the attachments.

IMail and EWSMail set an EmailCount at the Batch level and at the following Document level variables for each email message accepted (Done): TYPE="Document", Message ID, Subject, From, To, DateSent, Priority, Body=User.

IMail also sets the following Page level variables for each attachment: TYPE="Other", IMAGEFILE=*attachment filename*.

When you do not use the `ex_EMLOption` action, EWSMail sets the following Page level variables for each attachment: TYPE="Other", IMAGEFILE=*attachment filename in batch* ATTACHNAME=*original attachment filename*.

When you use the `ex_EMLOption` action for EWSMail, pages for attachments are not created, and variables for those attachments are not set.

Email Input actions

Use the actions in IMail.RRX when your mail server uses the Internet Message Access Protocol (IMAP). These include Microsoft Exchange Server, Novell GroupWise, and others.

Use the EWSMail Input actions when your mail server is an Microsoft Exchange mail server, which is configured to allow Exchange Web Service access.

Before you configure these actions, record the appropriate values for your system and have them available during the configuration process.

Table 1. Required Email Input actions parameter settings

Action	IMail.RXX	EWSMail.RXX
EWS Version		Version of Microsoft Exchange to use for EWSMail Input actions
Logon	URL of mail server, username, and password of the mail account	URL of mail server, username, and password of the mail account
Scan	None	None
Logout	None	None
Types	List of image files extensions to import	List of image files extensions to import
Wait Time	Maximum number of seconds to wait for input emails for a single batch	Maximum number of seconds to wait for input emails for a single batch
Abort Time	Number of seconds to wait before returning an abort	Number of seconds to wait before returning an abort
Max Docs	Maximum number of emails in each batch	Maximum number of emails in each batch
Done Folder	Name of folder into which successfully imported emails are stored	Name of folder into which successfully imported emails are stored
Problem Folder	Name of folder into which unsuccessfully imported emails are stored	Name of folder into which unsuccessfully imported emails are stored
EML Option		Optional for EWSMail Input actions: Create a one page document that contains the email and attachment in an .eml file. No attachment pages are created.

Email Sending actions

This table describes the parameters that are required for Email Sending actions. Before you configure these actions, record the appropriate values for your system and have them available during the configuration process.

Table 2. Required Email Sending actions parameter settings

Action	Description
Send Email	None
Set Attachment	Pathname and file name of the file you want to attach to the current email. Smart parameters are supported.

Action	Description
Set Blind Carbon Copy Receipts	Email addresses that receive a copy of the email as a blind carbon copy. You can enter multiple email addresses separated by commas.
Set Carbon Copy Receipts	Email addresses that receive a copy of the email as a carbon copy. You can enter multiple email addresses separated by commas.
Set Mail Body	Email message text. Smart parameters are supported
Set Mail Server	IP or DNS address of the outgoing mail (SMTP) server
Set Recipients	Email addresses of the recipients of the email
Set Sender	Email address of the sender of the email
Set Subject	Subject line of the email. Smart parameters are supported.

Parent topic: [Email Connector actions](#)

Configuring Email Connector actions

Email Connector can actions to scan Email servers that support IMAP protocol or Exchange Web Service (EWS) for incoming email messages with attachments. You specify and export the attachments into a batch.

Procedure

To configure Email Connector actions:

1. For EWS Email servers, specify the version of the mail server you want to scan for attachments.
2. Add the Email actions for your Email server (IMail.RRX or EWSMail.RRX) to the Export rulesets.

Example

The following example describes a Use EWS ruleset that selects the version of the EWS Email server to use. The ruleset logs on to the mail server, scans the server for incoming mail with attachments and imports them into the batch.

The ruleset contains the Connect to EWS and Find Attachment rules. The Connect to EWS contains the Version and Logon functions and actions you must make the connection to the wanted version of the Email server. The Find Attachment rule contains the Scan function with actions that locate the specified attachments and import them into the batch.

Export EWS ruleset

- Connect to EWS rule
 - Version function
 - `ex_ews_version("1")`
 - Logon function
 - `ex_login("hostname", "username", "password")`
- Find Attachment rule
 - Scan function
 - `ex_types("tiff", "pdf")`
 - `ex_scan()`

Parent topic: [Email Connector actions](#)

Email Connector import examples

The Email Connector actions connect to the Email servers that support IMAP protocol or Exchange Web Service (EWS). These actions scan incoming email messages for attachments that contain the types of attachments that you specify and import the attachments into a batch.

Import from an IMAP Email server

The examples in the following tables show the sequence in which you must add the actions to the Export ruleset for the different Email servers.

Table 1. The sequence of actions to use for an IMAP Email server

Action	Description
<code>im_login("hostname", "userid, password")</code>	Log in to Email server for the specified host name.
<code>im_types("tif", "pdf")</code>	Specify the email attachment types that you want to import.
<code>im_done_folder("folder_name")</code>	Specify the destination IMAP folder for successfully imported email messages. If this Action is not called, the default folder named Done is used.
<code>im_problem_folder("folder_name")</code>	Specify the destination IMAP folder for unsuccessfully imported email messages. If this Action is not called, the default folder named Problem is used.
<code>im_scan()</code>	Scan the email messages in the Inbox for the specified types. Imports the selected emails and attachments into the batch.
<code>im_logout</code>	Disconnect from the Email server.

Import from an EWS Email server

Table 2. The sequence of actions to use for an EWS Email server

Action	Description
<code>ex_ews_version("1")</code>	Specify the version of the EWS Email server to use: <ul style="list-style-type: none">• Type 1 for Exchange 2007 SP1• Type 2 for Exchange 2010• If called by any other parameter, the latest known library is used. Currently .NET 3.5 library on Exchange 2010 If the action is not called at all, defaults to the latest version.
<code>ex_login("hostname", "userid, password")</code>	Log in to Email server for the specified host name.
<code>ex_types("tif", "pdf")</code>	Specify the email attachment types that you want to import.
<code>ex_done_folder("folder_name")</code>	Specify the destination EWS folder for successfully imported email messages. If this Action is not called, the default folder named Done is used.

Action	Description
<code>ex_problem_folder("folder_name")</code>	Specify the destination EWS folder for unsuccessfully imported email messages. If this Action is not called, the default folder named Problem is used.
<code>ex_scan()</code>	Scan the email messages in the Inbox for the specified types. Imports the selected emails and attachments into the batch.
<code>ex_logout</code>	Disconnect from the Email server.

Parent topic: [Email Connector actions](#)

Fax Connector actions

You can use Fax Connector actions to create Datacap document batches from incoming faxes. You can also send the contents of a document to a specified fax number.

The Datacap Connector for Fax actions do the following steps to create document batches from information that you receive in faxes.

- Set the name of the Fax Server and the user ID and Password to use to connect to the OpenTextFaxServer
- Specify the protocol to use to connect to the OpenTextFaxServer
- Define the amount time before you stop running a batch, polling interval time, and the server authentication method.
- Connect to the OpenTextFaxServer
- Configure the maximum number of faxes in a batch and whether to remove processed faxes from the server
- Import the faxes into the document batch from the OpenTextFaxServer
- [Fax Connector prerequisites](#)
To configure and run Fax Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 8.0, 8.0.1, and 9.0.
- [Fax Connector settings](#)
Record the system settings that you want to use to configure the Fax Connector actions and have these values available during the configuration process.
- [Configuring Fax Connector actions](#)
You must create an Export ruleset and configure its rules and functions with Fax Connector actions to import incoming faxes into a document batch.
- [Fax Connector import examples](#)
The Fax Connector Actions connect to the OpenTextFaxServer to import incoming faxes into document batches.

Parent topic: [Datacap Connector actions](#)

Fax Connector prerequisites

To configure and run Fax Connector actions, your environment must meet the hardware and software requirements for Datacap, Version 8.0, 8.0.1, and 9.0.

The following components must be installed and running on your system before you can use the Fax Connector actions:

- Datacap Version 8.0, 8.0.1, and 9.0 installed and running on either a single computer or a client/server installation.
- RightFax client installed on each computer where the rules run.

Parent topic: [Fax Connector actions](#)

Related information:

[Hardware and Software Requirements for IBM Datacap Version 9.0.1](#)

Fax Connector settings

Record the system settings that you want to use to configure the Fax Connector actions and have these values available during the configuration process.

This table describes the parameters that are required for the Datacap Connector for Fax actions.

Table 1. Required Fax Connector actions parameter settings

Action	Description
Set Server Name	Name of the OpenTextFaxServer
Set User ID	User ID used to log in to the OpenTextFaxServer
Set Password	Password that is used to log in to the OpenTextFaxServer
Set Windows Authentication	Whether to use Windows Authentication to connect to the OpenTextFaxServer
Set Protocol	Protocol that is used to connect to the OpenTextFaxServer
Set Polling Interval	Number of milliseconds to wait before fax polling from the OpenTextFaxServer resumes
Set Abort Timeout	Number of seconds to wait before a batch run is stopped
Set Max Number of Faxes	Maximum number of faxes in each batch
Set Fax Removal After Import	Whether to remove processed faxes from the Fax Server, must be set to true so that new faxes are imported each time. If this action is not called or set to false, the Import Faxes action imports the same faxes over and over.
Import Faxes	Import the faxes from the OpenTextFaxServer into the document batch
Connect	Connect to the OpenTextFaxServer
Send Faxes	Fax the contents of the document or page to the specified Fax number
Disconnect	Disconnect from the OpenTextFaxServer

Parent topic: [Fax Connector actions](#)

Configuring Fax Connector actions

You must create an Export ruleset and configure its rules and functions with Fax Connector actions to import incoming faxes into a document batch.

Procedure

To configure Fax Connector actions:

1. Specify the name of the Fax Server from which you want import faxes.
2. Add the OpenTextFaxServer.RRX actions to the Export rulesets.

Example

The following example describes an Export Fax ruleset that selects the OpenTextFaxServer to use. The ruleset logs on to the server, and imports the faxes from the server into the batch.

The ruleset contains the Connect to Fax Server and Import Fax rules. The Connect to Fax rule contains the Server name, Logon, Protocol, and Connect functions and the actions that make the connection to the Fax server. The Import Fax rule contains the Import function with actions that locate the specified attachments and import them into the batch.

Export Fax ruleset

- Connect to Fax Server rule
 - Server name function
 - SetServerName("myserver")
 - Logon function
 - SetUserID("myuserID")
 - SetUserPassword("myPassword")
 - Protocol function
 - SetProtocol("4")
 - Connect function
 - Connect()
- Import Fax
 - Import function
 - ImportFaxes()

Parent topic: [Fax Connector actions](#)

Fax Connector import examples

The Fax Connector Actions connect to the OpenTextFaxServer to import incoming faxes into document batches.

Import faxes from a Fax server

The examples in the following tables show the sequence in which you must add the actions to the Export ruleset.

Table 1. The sequence of actions to import faxes

Action	Description
SetServerName("hostname")	Set the name of the OpenTextFaxServer to which you want to connect.
SetUserID("userid")	Set the user ID to use to log in to the OpenTextFaxServer.
SetUserPassword("Password")	Set the Password for the user ID to use to log in to the OpenTextFaxServer.

Action	Description
<code>SetProtocol("4")</code>	Specify the protocol to use to connect to the OpenTextFaxServer. The default value is "4" for TCP/IP. The other valid values are. <ul style="list-style-type: none"> • "1" - Named Pipes • "2" - IPXOS2 • "3" - SPX • "5" - IPX • "6" - SecTCPIP • "7" - SecSpx
<code>SetPollingInterval("2000")</code>	Set the amount of time, in milliseconds, to wait before you poll the OpenTextFaxServer for faxes again. The default value is "2000" (2 seconds).
<code>Connect()</code>	Connect to the Fax Server.
<code>SetFaxRemovalAfterImport(True)</code>	Removes faxes after they are imported to enable the new faxes to be imported when they are ready.
<code>ImportFaxes()</code>	Import the faxes from the OpenTextFaxServer and store them in a document in a batch.
<code>Disconnect()</code>	Close the connection to the OpenTextFaxServer.

Parent topic: [Fax Connector actions](#)

Box Connector actions

You can use the DatacapBOX connector actions to move data between your IBM® Datacap system and Box.com.

Specifically, you can export the following types of DCO objects to Box as files: pages and document-level PDF files. You can import Box files as one of the following types of DCO objects: document-level pages and batch-level pages. The word *Box* as used here refers to your account on Box.com.

- [Configuring Box Connector actions](#)
Some configuration is required to make the Box-related actions and rulesets visible in the user interface of Datacap programs and ready for use.
- [Box Connector settings](#)
Record the system settings that you want to use to configure the Box Connector actions.
- [Box Connector upload examples](#)
A ruleset for exporting data to Box performs three main functions: establishing the authentication token; specifying upload settings; and initiating the export process. The initiation of the export process (with the Upload action) must be the last action in the ruleset sequence.

Parent topic: [Datacap Connector actions](#)

Configuring Box Connector actions

Some configuration is required to make the Box-related actions and rulesets visible in the user interface of Datacap programs and ready for use.

Procedure

To use a Box-related ruleset in your application:

1. Record the Box-related settings and parameter values that you need for Box configuration. For more information, see [Box Connector settings](#).
2. If your machine is isolated from the internet and cannot directly communicate with Box, configure your machine to use a proxy server. Because the Box rulesets do not support proxy servers, external configuration of a proxy service is necessary to support network traffic to and from Box. For example, you might configure your Windows machine to use the WinHTTP proxy service to direct traffic to your HTTP proxy server. Regardless of the way that you implement the proxy server, the server must handle the Microsoft HTTP API functions that Datacap uses for the Box rulesets.
3. Copy the files from the directory <target installation path>\RRS\Box to the rules directory of your application. For example, you might copy files from C:\Datacap\RRS\Box to C:\Datacap\APT\dco_APT\rules. The Box directory comes installed with Datacap.
4. In your application's rules directory, edit the DatacapBOX.dll.config file to substitute the appropriate OAuth parameter values for occurrences of the following text:
 - [your client_id]
 - [your client_secret]
 - [your redirect_uri]
5. Use Datacap FastDoc or Datacap Studio to configure a Box-related ruleset. Two such rulesets come installed with Datacap: *Export to Box* and *Import from Box*. For information about creating a custom export ruleset, see [Box Connector upload examples](#) and [Configuring export options](#).
6. Configure Rulerunner to run your rulesets. In particular, specify the appropriate number of Rulerunner threads. For information about the appropriate thread number for your ruleset, see [Export object](#) (for export rulesets) or [Import object](#) (for import rulesets). For information about configuring Rulerunner, see [Configuring Rulerunner to run tasks](#).

Parent topic: [Box Connector actions](#)

Box Connector settings

Record the system settings that you want to use to configure the Box Connector actions.

Datacap connects to Box by using OAuth, a standard for authenticating and authorizing users. Specifically, the following OAuth parameters are used:

- Client ID
- Client secret
- Redirect URL

Record your values for these parameters when you create a Box app. For information about creating a Box app, see <https://app.box.com/developers/services>.

Table 1. Export settings

Setting	Description
Batch subfolder creation	Whether to create batch subfolders in the Box target folder. A subfolder has the same name as the batch and contains the exported files for the batch.

Setting	Description
Existing file handling	How the export process responds to the following circumstance: an existing version of a file to be exported exists in the Box target folder. Here are the possible responses: <ul style="list-style-type: none"> • The new file version is added. • The new file version is added, and the last version is deleted. • The export process fails and does not continue.
Existing metadata handling	How the export process handles the existing Box metadata for a file to be exported. This existing metadata is for the existing Box versions of a file in the Box target folder with the same name as the exported file. One set of metadata is associated with all versions of the same Box file. Here are the possibilities: <ul style="list-style-type: none"> • The metadata for the exported Box file includes the newly exported metadata only. • The metadata for the exported Box file includes the newly exported metadata and the previous Box metadata for the file before the export. If the same field exists in both sets of metadata, the value from the newly exported metadata is used.
Metadata object scope	Whether the exported metadata for a page includes document and batch data in addition to page data.
Metadata type	Whether the exported metadata includes fields, DCO variables, or both fields and variables.
PDF conversion	Whether documents are to be converted to multipage PDFs as part of the export.
Target Folder	The Box folder (if any) in which the export process places exported files. You can specify the target folder as a path. If the specified target folder does not exist, the export process creates it. By default, the target folder is your root Box folder.

Table 2. Import settings

Setting	Description
Backup folder	The Box folder (if any) to which the import process moves imported Box files. As part of this file relocation, imported Box files are deleted from the source folder. This deletion prevents files from being imported more than once.
Import as document type	The type of document that the import process creates (if any) to hold imported files as child pages.
Import limit	The maximum number of files that the import process imports from Box.
Imported file extensions	A comma-delimited list of file extensions that the import process uses as a criterion for determining which files to import from Box. A file must have one of the specified extensions to be imported.
Source folder	The Box source folder from which the import process imports files.

Parent topic: [Box Connector actions](#)

Box Connector upload examples

A ruleset for exporting data to Box performs three main functions: establishing the authentication token; specifying upload settings; and initiating the export process. The initiation of the export process (with the Upload action) must be the last action in the ruleset sequence.

Example

Here is an example ruleset:

Action	Description
TargetFolder("Datacap")	Specifies the Box target folder as Datacap. For more information, see TargetFolder .
CreateBatchSubfolder(True)	Indicates that batch subfolders are to be created in the Box target folder. For more information, see CreateBatchSubfolder .
AddParentDataToPageMetadata(True)	Indicates that the parent batch and parent document data is to be included in the exported metadata for a page. For more information, see AddParentDataToPageMetadata .
FailIfFileExists(False)	Indicates that the export process does not fail in the following circumstance: an existing file in the Box target folder has the same name as a file to be exported. For more information, see FailIfFileExists .
OverwriteExistingFiles(False)	For a file to be exported, indicates that the latest existing version of that file in Box is not to be deleted. For more information, see OverwriteExistingFiles .
ReplaceMetadata(False)	Indicates that the newly exported metadata for a file is to be merged with any existing Box metadata for that file. For more information, see ReplaceMetadata .
FieldsAreMetadata(True)	Indicates that field values are to be included in the exported metadata. For more information, see FieldsAreMetadata .
DCOVarsAreMetadata(False)	Indicates that DCO variables are not to be included in the exported metadata. For more information, see DCOVarsAreMetadata .
DocumentsToPDF(False)	Indicates that pages (as opposed to PDF files) are to be exported to Box. For more information, see DocumentsToPDF .
ProcessChildren(True)	Indicates that all child objects of the current DCO object are to be exported. For more information, see ProcessChildren .
Upload()	Initiates the export process. For more information, see Upload .

Parent topic: [Box Connector actions](#)

Connector actions log files

A log file contains the results of calling the action and explains why a document was not created when you upload documents into a repository. The name of the log file is based on the name of the task, for example export_rss.log.

After the documents are uploaded into a repository, the following file for that repository is created in the batch directory. This file contains the names of the files that were uploaded.

Repository	File name
------------	-----------

Repository	File name
IBM® CMIS server	CMIS_Uploaded.xml
IBM Content Manager	IBMCM_Uploaded.xml
IBM FileNet® Content Engine	FNP8_Uploaded.xml
Documentum Docbase	DM_Uploaded.xml
SharePoint library	SP_Uploaded.xml
FileNet Image Services	FileNetIDM_Uploaded.xml
Email Input	EWSmail_Uploaded.xml
Email Send	Email_Uploaded.xml
Open Text Fax Server	OpenTextFaxServer_Uploaded.xml

Parent topic: [Datacap Connector actions](#)

Viewing action details

Datacap Studio provides help topics with detailed information for all of the connector actions. The topics include the action library name, description, parameters, DCO level, returns, and examples for each action.

About this task

You can refer to these descriptions when you configure the parameters of the Connector actions. These actions are specific to the entity into which you are uploading files. When you configure actions for an application, you must set the actions for the entity in the sequence in which they are presented in the examples.

Procedure

To view connector action descriptions:

1. In Datacap Studio, click the New Actions tab.
2. Select the Action library.

Table 1. Action library names by connector

Connector	Action Library
IBM® Content Manager Connector	IBMCM
FileNet® P8 Connector	FileNetP8
Documentum Connector	Documentum
SharePoint Connector	SPEXport
FileNet Image Services Connector	FileNetIDM
eMail and eDoc Connector	IMail, EWSMail, EMail
Fax Connector	OpenTextFaxServer
Box Connector	DatacapBOX

3. Right-click on the action for which you want detailed information and select Information.

Parent topic: [Datacap Connector actions](#)

TravelDocs: Exporting data to a database

You can update the TravelDocs application to export data from each rental agreement page to an export database.

- [Configuring the export database](#)
You must use Datacap Application Manager to configure the export database.
- [Creating the ExportDB ruleset](#)
You use Datacap Studio to create the ExportDB ruleset. Also, you access the Datacap Studio actions library to add functions to the ruleset.
- [Adding the ExportDB ruleset to the Export task profile](#)
After you create the ExportDB ruleset, you must add it to the Export task profile.
- [Attaching the Export Rental Agreement Data rule to the rental agreement page](#)
You use Datacap Studio to attach the Export Rental Agreement Data rule to the rental agreement page.
- [Running a batch through the workflow](#)
After you create and configure the ExportDB ruleset and attach the data rule to the rental agreement page. You can run a batch and confirm that the export task is operational.

Parent topic: [Data export](#)

Configuring the export database

You must use Datacap Application Manager to configure the export database.

Procedure

To configure the export database:

1. In the Start menu click IBM Datacap Services > Datacap > Datacap Datacap Application Manager.
2. Select the TravelDocs application from the list on the left.
3. Click Browse [...] beside the Export database field.
4. In the Database type field, select Microsoft Access. Then, in the Database field, select the file C:\Datacap\TravelDocs\TravelDocsExport.mdb. Database authentication is not needed.
5. Click OK and then close the Datacap Application Manager window.

Parent topic: [TravelDocs: Exporting data to a database](#)

Related information:
[Application Manager](#)

Creating the ExportDB ruleset

You use Datacap Studio to create the ExportDB ruleset. Also, you access the Datacap Studio actions library to add functions to the ruleset.

Procedure

To create the ExportDB ruleset:

1. In the Rulesets pane, right-click the TravelDocs node and choose Add Ruleset.
2. Rename the new ruleset from Ruleset1 to ExportDB.
3. Rename the default rule from Rule1 to Export Rental Agreement Data.
4. Rename the default function from Function1 to ExportDB.

- Click the Actions library tab and expand the ExportDB library.
- Select and add each of the following actions that are shown in the following table to the Export Data function by clicking Add to function. Then, set the action parameters as shown in the following table.

Action	Parameter
ExportOpenConnection	@APPVAR(*)/exportdb:cs)
SetTableName	Rental_Agreement
ExportBatchIDToColumn	BatchID
ExportFieldToColumn	Pickup_Date,Pickup_Date
ExportFieldToColumn	Pickup_Location,Pickup_Location
ExportFieldToColumn	Return_Date,Return_Date
ExportFieldToColumn	Return_Location,Return_Location
ExportFieldToColumn	Car_Type,Car_Type
ExportFieldToColumn	Options,Options
ExportFieldToColumn	Total_Cost>Total_Cost
AddRecord	
ExportCloseConnection	

- In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and choose Publish ruleset.

Parent topic: [TravelDocs: Exporting data to a database](#)

Adding the ExportDB ruleset to the Export task profile

After you create the ExportDB ruleset, you must add it to the Export task profile.

Procedure

To add the ExportDB ruleset to the Export task profile:

- In the Rulesets pane, select the ExportDB ruleset.
- Click the Task profiles tab and click Lock/Unlock task profiles.
- Select the Export task profile and click the Add ruleset to profile button in the Task profiles pane.
- Expand the Export task profile and make sure the ExportDB ruleset is listed.
- In the Task profiles pane, click Save. Then, click Lock/Unlock task profiles.

Parent topic: [TravelDocs: Exporting data to a database](#)

Attaching the Export Rental Agreement Data rule to the rental agreement page

You use Datacap Studio to attach the Export Rental Agreement Data rule to the rental agreement page.

Procedure

To attach the Export Rental Agreement Data rule to the rental agreement page:

1. In the Document hierarchy pane, click Lock DCO for editing.
2. Expand the document hierarchy so the Rental_Agreement page is visible. Then, select the Rental_Agreement page.
3. In the Rulesets pane, select the Export Rental Agreement Data rule and click Add to DCO.
4. With the Export Rental Agreement Data rule still highlighted, click Sync DCO view with Ruleset view. Make sure that the new rule is now included in the Open element of the rental agreement page.
5. In the Document hierarchy pane, click Save, then click Unlock DCO.

Parent topic: [TravelDocs: Exporting data to a database](#)

Running a batch through the workflow

After you create and configure the ExportDB ruleset and attach the data rule to the rental agreement page. You can run a batch and confirm that the export task is operational.

Procedure

To run a batch through the workflow:

1. Use the Connection wizard to reopen the TravelDocs application. Opening the Connection wizard, forces Datacap Studio to reload the information from the application configuration (.app) file. If you do not open the wizard, the export database connection string might not be in the cached copy of the .app file. The new ruleset might fail.
2. In Datacap Studio, click the Test tab.
3. In the Rulesets pane, expand the ExportDB ruleset. Then, right-click the Export Rental Agreement Data rule and choose Set breakpoint. The action stops when Datacap reaches the rule.
4. In the Workflow pane, select the VScan task profile under Main Job.
5. Click New to start a new batch.
6. Click Process rules for target object and Advance to move the batch through the VScan, PageID, Profiler, Verify, and Export tasks. The action stops at the breakpoint.
7. Click Step in to single-step into the function and start running the actions. As each line completes, ensure that there is a check mark beside the action, which indicates that the action returned `True`.
Tip: If `ExportOpenConnection` fails (as indicated by a ! beside the action), ensure that you set up the export database correctly. And that you added the connection string to the Datacap Application Manager. Then, use the Connection wizard to reopen the TravelDocs application.
8. Click Process rules for target object to resume normal execution. You must click Process rules for target object each time you press the Export Rental Agreement Data rule (for each rental agreement page). Then, click Advance.
9. Open the file `C:\Datacap\TravelDocs\TravelDocsExport.mdb` and review the exported data in the Rental_Agreement table.

Parent topic: [TravelDocs: Exporting data to a database](#)

TravelDocs: Exporting data to an XML file

You can update the TravelDocs application to export data from each rental agreement page to an XML file. If you want to export data from the other pages, you must have a separate rule for each page type.

- [Creating the ExportXML ruleset](#)
You use Datacap Studio to create the ExportXML ruleset. Also, the ruleset requires three rules.
- [Adding the Export XML ruleset to the Export task profile](#)
After you create the Export XML ruleset, you must add it to the Export task profile.

- [Attaching the Export XML rules to the document hierarchy](#)
After you add the ruleset to the task profile, you must attach the rules to the document hierarchy.
- [Running a batch through the workflow](#)
After you create and configure the ExportXML ruleset, and attach the required rules to the document hierarchy. You can run a batch and confirm that the export task is operational.

Parent topic: [Data export](#)

Creating the ExportXML ruleset

You use Datacap Studio to create the ExportXML ruleset. Also, the ruleset requires three rules.

About this task

Three separate rules are required.

- One rule that is attached to the Open element of the batch to set the XML export path and file name.
- One rule that is attached to the rental agreement page that writes the data for the current page.
- One rule that is attached to the Close element of the batch to save the XML file.

Procedure

To create the ExportXML ruleset:

1. In the Rulesets pane, right-click the TravelDocs node and choose Add Ruleset.
2. Rename the new ruleset from Ruleset1 to Export XML.
3. Rename the default rule from Rule1 to Open XML File.
4. Rename the default function from Function1 to Open XML.
5. Click the Actions library tab and expand the Export XML library.
6. Select and add each of the following actions that are shown in the following table to the Open XML function by clicking Add to function. Then, set the action parameters as shown in the following table.

Action	Parameter
xml_SetExportPath	@APPPATH(export)
xml_SetFileName	@BatchID

Attention: @APPPATH(export) is a smart parameter that gets the export path from the application configuration file. @BatchID is a smart parameter that returns the current batch ID.

7. Right-click the ExportXML ruleset and choose Add Rule.
8. Rename the new rule from Rule1 to Export Rental Agreement XML.
9. Rename the default function from Function1 to Export XML.
10. Select and add each of the following actions that are shown in the following table to the Export XML function by clicking Add to function. Then, set the action parameters as shown in the following table.

Action	Parameter
xml_NewNode	@ID,Rental_Agreements
xml_NewNode	Pickup_Date,@ID
xml_SetNodeValue	Pickup_Date, @P\Pickup_Date
xml_NewNode	Pickup_Location,@ID
xml_SetNodeValue	Pickup_Location, @P\Pickup_Location
xml_NewNode	Return_Date,@ID

Action	Parameter
xml_SetNodeValue	Return_Date, @P\Return_Date
xml_NewNode	Return_Location,@ID
xml_SetNodeValue	Return_Location, @P\Return_Location
xml_NewNode	Car_Type,@ID
xml_SetNodeValue	Car_Type, @P\Car_Type
xml_NewNode	Options,@ID
xml_SetNodeValue	Options, @P\Options
xml_NewNode	Total_Cost,@ID
xml_SetNodeValue	Total_Cost, @P\Total_Cost

Attention: @ID gets the ID of the current object. @P\ gets the value of the specified field on the current page.

11. Right-click the ExportXML ruleset and choose Add Rule.
12. Rename the new rule from Rule1 to Close XML File.
13. Rename the default function from Function1 to Close XML.
14. Select and add the action that is shown in the following table to the Close XML function by clicking Add to function. This action has no parameter.

Action	Parameter
xml_SaveFile	

15. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and choose Publish ruleset. The finished ruleset looks like the following example.

Parent topic: [TravelDocs: Exporting data to an XML file](#)

Adding the Export XML ruleset to the Export task profile

After you create the Export XML ruleset, you must add it to the Export task profile.

Procedure

To add the Export XML ruleset to the Export task profile:

1. In the Rulesets pane, select the Export XML ruleset.
2. Click the Task profiles tab and click Lock/Unlock task profiles.
3. Select the Export task profile and click Add ruleset to profile in the Task profiles pane.
4. Expand the Export task profile and ensure that the Export XML ruleset is displayed.
5. In the Task profiles pane, click Save. Then, click Lock/Unlock task profiles.

Parent topic: [TravelDocs: Exporting data to an XML file](#)

Attaching the Export XML rules to the document hierarchy

After you add the ruleset to the task profile, you must attach the rules to the document hierarchy.

Procedure

To attach the Export XML rules to the document hierarchy:

1. In the Document hierarchy pane, click Lock DCO for editing.
2. Expand the batch and select the Open element of the batch.
3. In the Rulesets pane, select the Open XML File rule and click Add to DCO.
4. Select the Close element of the batch.
5. In the Rulesets pane, select the Close XML File rule and click Add to DCO.
6. In the Document hierarchy pane, expand the Car_Rental document node and select the Rental_Agreement page.
7. In the Rulesets pane, select the Export Rental Agreement XML rule and click Add to DCO.
8. Select the Open XML File rule, click Sync DCO view with Ruleset view. Make sure that the rule is now included in the Open element of the batch.
9. Repeat for the Export Rental Agreement XML and Close XML File rules.
10. In the Document hierarchy pane, click Save, then click Unlock DCO.

Parent topic: [TravelDocs: Exporting data to an XML file](#)

Running a batch through the workflow

After you create and configure the ExportXML ruleset, and attach the required rules to the document hierarchy. You can run a batch and confirm that the export task is operational.

Procedure

To run a batch through the workflow:

1. Click the Datacap Studio Test tab.
2. In the Breakpoints pane, click Remove all breakpoints.
3. In the Workflow pane, select the VScan task profile under Main Job.
4. Click New to start a new batch.
5. Click Process rules for target object and Advance to move the batch through the entire workflow
6. Open the file C:\Datacap\TravelDocs\export\batch_Identifier.xml and review the exported XML data.

```
<?xml version='1.0' ?>
<Rental_Agreements>
<TM000001>
<Pickup_Date>Tues, Dec 7, 2010</Pickup_Date>
<Pickup_Location>Boston (BOS)</Pickup_Location>
<Return_Date>Fri, Dec 10, 2010</Return_Date>
<Return_Location>Boston (BOS)</Return_Location>
<Car_Type>Compact</Car_Type>
<Options>Fuel Service</Options>
<Total_Cost>345.70</Total_Cost>
</TM000001>
<TM000003>
<Pickup_Date>Mon, Dec 6, 2010</Pickup_Date>
<Pickup_Location>San Francisco (SFO)</Pickup_Location>
<Return_Date>Fri, Dec 10, 2010</Return_Date>
<Return_Location>San Francisco (SFO)</Return_Location>
<Car_Type>SUV</Car_Type>
<Options>Child Seat</Options>
<Total_Cost>489.31</Total_Cost>
</TM000003>
<TM000004>
<Pickup_Date>Mon, Dec 13, 2010</Pickup_Date>
<Pickup_Location>Newark (EWR)</Pickup_Location>
<Return_Date>Thur, Dec 16, 2010</Return_Date>
<Return_Location>Newark (EWR)</Return_Location>
```

```
<Car_Type>Luxury</Car_Type>
<Options>Navigation System Child Seat Fuel Service</Options>
<Total_Cost>387.40</Total_Cost>
</TM000004>
</Rental_Agreements>
```

Parent topic: [TravelDocs: Exporting data to an XML file](#)

Application Debugging

Application debugging requires that you review two runtime log files, which are the Rulerunner Service (RRS) log and the task log. The RRS log provides detailed information about each action and is most helpful to application developers. The task log documents internal calls and is used mostly by IBM software support.

Datacap Studio includes integrated debugging functionality through which you can control the execution environment and monitor your application at runtime.

- [Datacap log files](#)
Datacap generates two types of log files during task execution.
- [Debug your application from the Datacap Studio Test tab](#)
You can run an application from Datacap Studio to monitor the application during execution to determine whether the rules are running as you expect. The Datacap Studio Test tab includes debugging features.

Parent topic: [Datacap application development](#)

Datacap log files

Datacap generates two types of log files during task execution.

- Rulerunner Service (RRS) log files include detailed information about each action as it runs.
- Task log files document mostly internal calls and is most helpful to IBM Software support.

Additionally, Report Viewer and Rulerunner can generate their own log files. Rulerunner logging is discussed in the topic [Rulerunner logging](#).

- [Enable logging for Datacap Web Client tasks](#)
To enable logging for a web task, you must configure that task in the Datacap Web Client.
- [Rulerunner Service \(RRS\) log files](#)
As Rulerunner runs each action, it writes detailed logging information to a Rulerunner Service (RRS) log file (*task_rrs.log*). Rulerunner also generates an RRS log file whenever you run a task from Datacap Studio.
- [Set Rulerunner logging by application and task](#)
You can set Rulerunner logging by application and by task when you add string values to the registry.
- [Task log files](#)
Logging is enabled by default for all tasks except VScan and tasks that you start from Datacap Studio.

Parent topic: [Application Debugging](#)

Enable logging for Datacap Web Client tasks

To enable logging for a web task, you must configure that task in the Datacap Web Client.

To enable logging from web client tasks:

1. In the Datacap Web Client, click the Administrator tab.
 2. On the Administrator tab, click Workflow.
 3. Expand the job that contains the task for which you want to enable logging, and select the task.
 4. Click Setup in the Selected task details pane.
 5. In the Rulerunner service log field, enter one of the values, as required.
- Tip: RRS logging is only useful for tasks that run rules. If your web client is not associated with a task profile, an RRS log file is not generated.

Rulerunner Manager Service log setting	Result
0 or 1	No RRS log file
2	RRS log file with action logging but no action parameters displayed
3 or 4	RRS log file with action logging and action parameters displayed
5 or higher	RRS log file with action logging and complete DCO navigation

In most situations, a setting of 3 provides enough information to help you debug rule-related issues.

Parent topic: [Datacap log files](#)

Rulerunner Service (RRS) log files

As Rulerunner runs each action, it writes detailed logging information to a Rulerunner Service (RRS) log file (*task_rrs.log*). Rulerunner also generates an RRS log file whenever you run a task from Datacap Studio.

If you want to generate an RRS log file for tasks that you run from the Datacap Web Client or for Datacap Desktop tasks, complete the following steps.

1. Start Datacap Rulerunner Manager.
2. Click the Logging tab.
3. Click the RRS log tab and select the logging options that you want.

In the Datacap Web Client, each task generates its own Rulerunner Service log file. The most recent TravelDocs batches folder contains a log file for each of the task profiles in the Main Job workflow.

Each log file contains detailed descriptions of the actions that are run by the task profile and is useful for application troubleshooting.

Example 1

Here is the *vscan_rrs.log* entry that shows execution of the SetSourceDirectory action in the VScan rule set:

```
[1] action SetSourceDirectory (bool=false,bool=true,str="@APPPATH(vscanimagedir)")
[2] 1 Smart Parameter element found
[3] Parsing Smart Parameter element {0} value: "@APPPATH(vscanimagedir)"
[4] @APPPATH key root value: 'vscanimagedir'
[5] @APPPATH looking for workflow key: '*/dco_TravelDocs/vscanimagedir'
[6] @APPPATH workflow key found: 'C:\Datacap\TravelDocs\images'
[7] Smart Parameter return value: 'C:\Datacap\TravelDocs\images'
[8] looking for:C:\Datacap\TravelDocs\images
[9] Action changes: Directory with source images: C:\Datacap\TravelDocs\images
[10] result 0[x0] = true
[11] action returned true
[12] execute statement On Action True
```

```

[13]     executing code:
[14]     Call OnActionEnd()
[15] /execute statement On Action True
[16] /action
[17] execute statement On Action Start
[18]     executing code:
[19]     Call OnActionStart()
[20] /execute statement On Action Start

```

By looking through the Rulerunner Service log file, you can see precisely how Rulerunner interprets and runs each action. In the SetSourceDirectory Example 1, Rulerunner:

- Identifies the @APPPATH(vscanimagedir) parameter as a smart parameter [line 2]
- Identifies the key value as vscanimagedir [line 4)]
- Looks up the specified key value in the application configuration [line 5]
- Retrieves the value C:\Datacap\TravelDocs\images [line 6]
- Sets the image source directory to the specified location [line 9]

Example 2

In the previous example, the action that is executed successfully and returned true [line 11]. In this next example, you will introduce an invalid key name in the action parameter:

```
SetSourceDir("@APPPATH(imagedir)") <--"imageddir" is not a valid key
```

This time the batch aborts. If you run the task from the Datacap Studio, you see an error message. In this case, you can look at the end of the log file to determine the cause of the error [line 6].

```

[1] action SetSourceDirectory (bool=false,bool=false,str="@APPPATH(imagedir)")
[2] 1 Smart Parameter element found
[3] Parsing Smart Parameter element {0} value: "@APPPATH(imagedir)"
[4] @APPPATH key root value: 'imagedir'
[5] @APPPATH looking for workflow key: '*/dco_TravelDocs/imagedir'
[6]     @APPPATH workflow key not found. <--Key not found
[7] @APPPATH looking for appname key: '*/dco_TravelDocs/imagedir'
[8]     @APPPATH appname key not found.
[9] @APPPATH looking for general key: '*/imagedir'
[10]    @APPPATH general key not found.
[11] Smart Parameter return value: ''
[12] looking for:@APPPATH(imagedir)
[13] Error: Folder '@APPPATH(imagedir)' does not exist <--Folder does not exist
[14] Error (385875969=hex:17000001). In CIMainAlgorithm::execute4DCO: Aborting:
Action [SetSourceDirectory]
    requested abort [api source:]
[15] EXCEPTION: code="385875969" msg="Aborting: Action [SetSourceDirectory]
    requested abort" loc="CIMainAlgorithm::execute4DCO" API=""
[16] execute statement On Process End
[17] executing code:
[18] Quit()

```

Parent topic: [Datacap log files](#)

Set Rulerunner logging by application and task

You can set Rulerunner logging by application and by task when you add string values to the registry.

To enable RRS logging by application name, add a String Value with the Value name `app_filter` to the registry. Then, enter the application name in the Value data field. For example, if you want to enable logging for the APT application tasks, add the `app_filter` value name and enter `APT` in the Value data field.

To enable RRS logging by task profile, add a String Value with the Value name `tprofile_filter` to the registry. Then, enter the task name in the Value data field. For example, if you want to enable logging for the VScan task, add the `tprofile_filter` value name and enter `VScan` in the Value data field.

- For 32-bit (x86) OS, add the value in the following location.
 - `HKEY_LOCAL_MACHINE\SOFTWARE\Datcap\Rulerunner\RRSLog`
- For 64-bit (x64) OS, add the value in the following location.
 - `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Datcap\Rulerunner\RRSLog`

You can set logging for an application and a task profile by adding the `app_filter` and the `tprofile_filter` string values to the registry. For example, if you set the `app_filter` value to `APT` and the `tprofile_filter` value to `VScan`, RRS logging is enabled only when the VScan task profile in the APT application is run.

If the `app_filter` Value data field is empty, RRS logging is enabled for all applications. If the `tprofile_filter` Value data field is empty, RRS logging is enabled for all task profiles.

To confirm that the registry log filter is enabled, look in the `Rulerunner.log` file or look in the batch.

- The `Rulerunner.log` file indicates if application or task filtering is enabled for RRS logging.
- The batch has only the `.rrs` log files for the filtered task. For example, if you set the `tprofile_filter` value to `VScan`, then only the `VScan.rrs` logs are in the batch.

Note: You must restart the Rulerunner Service when you add a registry key log filter or change the value of a log filter.

Parent topic: [Datacap log files](#)

Task log files

Logging is enabled by default for all tasks except VScan and tasks that you start from Datacap Studio.

If you want to generate a task log for Datacap Web Client tasks, you must set the severity level to 1 or higher (on the scale of 0 to 9). The severity level and the options in the Datacap Web Client Task setup window determine how much information is written to the log file. (See [Enable logging for Datacap Web Client tasks](#)).

The task log file is saved in the batch folder.

The task log provides information that is typically most helpful to IBM® support because it documents mostly internal calls.

Parent topic: [Datacap log files](#)

Debug your application from the Datacap Studio Test tab

You can run an application from Datacap Studio to monitor the application during execution to determine whether the rules are running as you expect. The Datacap Studio Test tab includes debugging features.

- [Using breakpoints](#)
A breakpoint stops a task at a predetermined rule set, rule, or action, or stops the task when a task starts processing a specific document, page, or field.
- [Single-stepping through your code](#)
Single-stepping is useful to determine whether the functions and actions within a rule are operating as intended. As you step through each line, you can see the actions that are returned as `True` (check mark) and `False`. (exclamation point).
- [Examining log files from the Test tab](#)
You can access the Output tab in Datacap Studio to view the output that is written to different log files.

Using breakpoints

A breakpoint stops a task at a predetermined rule set, rule, or action, or stops the task when a task starts processing a specific document, page, or field.

- [Breakpoint types](#)
There are two types of breakpoints, both of which halt execution when Rulerunner encounters the specified element.
- [Setting breakpoints](#)
You can set breakpoints for a rule set, rule, function, action, document, page, or field.
- [Disable and clear breakpoints](#)
You can enable or disable individual breakpoints by selecting or clearing check boxes. The buttons on the left of the Breakpoints pane are options to enable, disable, or remove breakpoints.
- [Set generic breakpoints](#)
The Datacap Studio Test tab includes two controls that you can select to halt processing when any rule or action fails

Parent topic: [Debug your application from the Datacap Studio Test tab](#)

Breakpoint types

There are two types of breakpoints, both of which halt execution when Rulerunner encounters the specified element.

- **Breakpoints:** Halts execution when the Rulerunner execution manager encounters the specified element, regardless of context.
- **Full Breakpoints:** Halts execution when the Rulerunner execution manager encounters the specified element within the same context.

To understand the difference between a breakpoint and a full breakpoint, consider that the TravelDocs application includes two calls to `ExportCloseConnection()` (one call in the Export Rental Agreement Data rule and one call in the Export Other Close Database rule.

- If you set a breakpoint on the first instance of `ExportCloseConnection`, execution stops whenever `ExportCloseConnection` is called from anywhere in the application.
- If you set a full breakpoint on the first instance of `ExportCloseConnection`, execution stops only when `ExportCloseConnection` is called from the Export Data function in the Export Rental Agreement Data rule in the ExportDB rule set.

You can see the difference in the way the breakpoints are defined by selecting the Breakpoints tab, which displays all of the defined breakpoints.

Parent topic: [Using breakpoints](#)

Setting breakpoints

You can set breakpoints for a rule set, rule, function, action, document, page, or field.

About this task

- For rule sets and rules, you can set only breakpoints, not full breakpoints.

- The document, page, or field must exist in the runtime hierarchy before you can set a breakpoint on it.

Procedure

To set breakpoints:

1. Set a breakpoint on a rule set, rule, function, or action:
 - a. Go to the Rulesets pane on the Datacap Studio Test tab.
 - b. Right-click the item and select Set breakpoint or Set full breakpoint.
2. Set a breakpoint on a document, page, or field:
 - a. Go to the Runtime batch hierarchy pane on the Datacap Studio Test tab.
 - b. Right-click the item and select Set breakpoint or Set full breakpoint.

Parent topic: [Using breakpoints](#)

Disable and clear breakpoints

You can enable or disable individual breakpoints by selecting or clearing check boxes. The buttons on the left of the Breakpoints pane are options to enable, disable, or remove breakpoints.

The Breakpoints pane displays all of the defined breakpoints.

The check box to the left of each breakpoint indicates whether the breakpoint is enabled or disabled. By default, breakpoints are enabled when you add them.

Table 1. Breakpoints pane checkboxes

Button	Description
Enable all breakpoints button	Enables all the breakpoints that are displayed in the Breakpoints pane.
Disable all breakpoints button	Disables all the breakpoints that are displayed in the Breakpoints pane.
Remove selected breakpoints button	Removes the highlighted breakpoints. To select multiple breakpoints, hold down the Ctrl key.
Remove all breakpoints button	Removes all breakpoints from the Breakpoints pane.

Parent topic: [Using breakpoints](#)

Set generic breakpoints

The Datacap Studio Test tab includes two controls that you can select to halt processing when any rule or action fails

Control	Command	Description
A!	Stop on a failed action	Click this button to add a generic breakpoint that halts processing whenever an action fails.
R!	Stop on a failed rule	Click this button to add a generic breakpoint that halts processing whenever a rule fails.

Parent topic: [Using breakpoints](#)

Single-stepping through your code

Single-stepping is useful to determine whether the functions and actions within a rule are operating as intended. As you step through each line, you can see the actions that are returned as `True` (check mark) and `False`. (exclamation point).

If an action returns false, you can look in the batch log to see why the action returned false. (See [Examining log files from the Test tab](#).)

Remember:

- If all actions within a function return `True`, Datacap skips any remaining functions in the current rule.
- If an action returns `False`, Datacap skips any remaining actions within that function and starts the next function (if there is one).

The Test tab provides UI controls (enabled with tooltips) for stepping through code:

UI control	Description
Step in	Steps into the next line of code. If the next line calls a rule or function, Step in opens the rule or function and halts inside it. If the next line is an action, Step in opens the action and you must click it again to close the action. For example, if a process is halted at a function, use this control to step into a function and start each of its actions.
Step/Step over	Starts the next line of code and any lower-level functions and actions, and then stops. If the next line is an action, Step over works like Step in and opens the action. For example, if a process is halted at a function, use this control to start the function, including all of its actions.
Step out	Steps through the next line of code. If the next line is a rule or function, Step out works like Step over and starts any lower-level functions and actions. If the next line is an action, Step out starts and closes the action.

Parent topic: [Debug your application from the Datacap Studio Test tab](#)

Examining log files from the Test tab

You can access the Output tab in Datacap Studio to view the output that is written to different log files.

Procedure

1. Click the Output tab in the center pane of the Test tab.
2. If Batch log is not already selected, click the down-arrow and select it from the list of available logs.

The Output pane refreshes automatically whenever you stop at a breakpoint or single-step through a line of code, or when the current task profile completes, although you need to scroll to the bottom each time to see the latest messages. For example, if you select a Stop on a failed action (`a[@res="false"]`) you can see the action that returned `False`.

Parent topic: [Debug your application from the Datacap Studio Test tab](#)

Handling line item grids

The techniques that you implemented rely upon data at predictable locations on the page. When you receive an invoice, you do not know how many items the invoice might contain. There might be just one item, or there might be a hundred items, possibly spanning multiple pages. Datacap includes actions to handle line item grids. You define the region on the page that might contain line items and define the structure of one line item. Datacap can then scan the region and locate all of the individual line items.

About this task

Later, you can update the TravelDocs application to demonstrate various techniques that are related to line item grids, including recognition, validation, verification, and export.

- [Defining the document hierarchy for line item grids](#)
A line item grid is a structure of repeating items, each of which typically contains several fields. To set up the document hierarchy, you must define the region of the page that can include line items. In addition, you must define the structure of one line item in terms of its fields.
- [Rules to recognize line items](#)
Datacap includes actions that simplify the processing of an undefined number of line items.
- [Text matching to locate fields](#)
Frequently, a line item grid includes a total at the bottom. There might be other fields, like sales tax and shipping costs. Because you do not know in advance how many line items are in the grid, you cannot know where the other fields are located. Because the location of the other fields is unpredictable, you cannot use positional information to read these fields. Instead, you can use text matching to locate an adjacent label and then read the text beside the label.
- [Removing non-line items from the page data file](#)
The region that is defined for the line item grid might include fields that are not line items. For example, the `Total` field that you used for text matching is not a line item. Therefore, Datacap might create line items for items that are not line items.
- [Exporting data from a line item grid](#)
You can export line item grid data by using the export actions. However, since the data exists at different levels in the runtime hierarchy, you need different rules for each level.
- [TravelDocs: Adding new pages that contain line item grids](#)
In this example, you update the TravelDocs application to process the Meals and Other Charges pages of the Hotel document type. Both of these pages include line item grids of undefined length.
- [TravelDocs: Recognizing line item grid data](#)
Recognizing the data with a line item grid requires these rules. Each rule is attached to a different object in the document hierarchy.
- [TravelDocs: Validating line item grid data](#)
In the TravelDocs application, you add validations to the Other Charges page to confirm that the calculations on them are correct.
- [TravelDocs: Verifying the line item grid pages](#)
For demonstration purposes, use Datacap Desktop to verify the line item grid pages.
- [TravelDocs: Exporting line item grid data to a database](#)
You can update the application to export line item grid data from the Other Charges page to a table in the export database.

Parent topic: [Datacap application development](#)

Defining the document hierarchy for line item grids

A line item grid is a structure of repeating items, each of which typically contains several fields. To set up the document hierarchy, you must define the region of the page that can include line items. In addition, you must define the structure of one line item in terms of its fields.

Procedure

You can define only one line item in the document hierarchy. At run time, Datacap expands the runtime hierarchy as needed to accommodate however many line items it finds.

```
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_TM00
0001">          <--Page
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Grid
_region">          <--Grid region
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Line
_item0">          <--First line item
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Item
"> etc. </F>          <--Item field
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Desc
ription"> etc. </F>          <--Description field
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Cost
"> etc. </F>          <--Cost field
  </F>
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Line
_item1">          <--Second line item
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Item
"> etc. </F>          <--Item field
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Desc
ription"> etc. </F>          <--Description field
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Cost
"> etc. </F>          <--Cost field
  </F>
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Line
_item2">          <--Third line item
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Item
"> etc. </F>          <--Item field
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Desc
ription"> etc. </F>          <--Description field
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev104_Cost
"> etc. </F>          <--Cost field
  </F>
  etc.
```

Parent topic: [Handling line item grids](#)

Rules to recognize line items

Datacap includes actions that simplify the processing of an undefined number of line items.

Table 1. Actions that simplify line item processing

Library	Action	Description
Zones	ScanDetails	Searches a line item grid object and looks for line items. Assign this action to the grid region in the document hierarchy.
Zones	ScanLineItem	Searches a line item object and looks for fields. Assign this action to each line item in the document hierarchy.
Zones	PopulateZNLinesItemField	Populates the page data file with the recognized value in the zone for the current line item child field. Assign this action to each line item child field in the document hierarchy.

These three actions automate the process of reading line item grids. When you are assigning rules, the key is to make sure that the actions operate at the correct level in the document hierarchy.

Datacap runs each rule, first on the grid and then on each line item and field in the grid. As Datacap runs, it builds a data structure in memory. The data gets written to the page data file upon completion of the current task.

Table 2. Resulting data structures for selected actions

ScanDetails()	ScanLineItem()	PopulateZNLinesItemField()
Runs once only (at the grid level)	Runs once for each line item	Runs once for each field
Resulting data structure (in memory): Grid_region <ul style="list-style-type: none"> • Line_item0 • Line_item1 	Resulting data structure (in memory): Grid_region <ul style="list-style-type: none"> • Line_item0 <ul style="list-style-type: none"> ◦ Item ◦ Description ◦ Cost • Line_item1 <ul style="list-style-type: none"> ◦ Item ◦ Description ◦ Cost 	Resulting data structure (in memory): Grid_region <ul style="list-style-type: none"> • Line_item0 <ul style="list-style-type: none"> ◦ Item = 1176 ◦ Description = Widget ◦ Cost = \$6.95 • Line_item1 <ul style="list-style-type: none"> ◦ Item = 9122 ◦ Description = Widget ◦ Cost = \$8.25

In this way the three actions can read all the items in a grid of arbitrary length.

Parent topic: [Handling line item grids](#)

Text matching to locate fields

Frequently, a line item grid includes a total at the bottom. There might be other fields, like sales tax and shipping costs. Because you do not know in advance how many line items are in the grid, you cannot know where the other fields are located. Because the location of the other fields is unpredictable, you cannot use positional information to read these fields. Instead, you can use text matching to locate an adjacent label and then read the text beside the label.

Datacap provides many actions for locating text on a page. You can consider these actions in more detail in the topics about text matching. Typically, when you work with line item grids, you are searching for the last instance of a word or phrase like *Total* or *Sales Tax*. The Locate actions library has several useful actions, including the actions in the following table.

Library	Action	Description
Locate	FindLastRegEx	Locates the last occurrence of a word or phrase on the current page.
Locate	FindLastKeywordList	Locates the last occurrence of a word or phrase that is contained in the specified keyword file. A keyword file is a text file with a .key extension that contains a list of similar words and phrases; for example, <i>Sales tax</i> and <i>Tax</i> .
Locate	GoRightWord	Moves <i>n</i> words to the right of the location of a previously found word.
Locate	UpdateField	Updates the current field in the page data file with the value of the located word.

For detailed information on these and other actions in the Locate library, select the action on the Actions Library tab and click Display information.

The The Recognize Grid Total Field rule in the TravelDocs application locates the last instance of the word *Total* on the current page. The rule then moves one word to the right and confirms that the value is a currency value. Finally, the rule updates the current field in the page data file. The rule must be attached to the `Total` field in the document hierarchy.

Parent topic: [Handling line item grids](#)

Removing non-line items from the page data file

The region that is defined for the line item grid might include fields that are not line items. For example, the `Total` field that you used for text matching is not a line item. Therefore, Datacap might create line items for items that are not line items.

About this task

In the following invoice example, although there are only two line items on the invoice, Datacap created a third line item for the `Total` field.

Item	Descript	Cost
1176	Widget	\$6.95
9122	Widget	\$8.25
	Total	\$15.20

The following is the XML code for the invoice example.

```
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_TM00
0001">
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Grid
_region">
```

```

    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Line
_item0">    <!--First line item
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Item
"> etc.</F>    <!--Has the value "1176"
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Desc
ription"> etc.</F>    <!--Has the value "Widget"
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Cost
"> etc.</F>    <!--Has the value "6.95"
    </F>
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Line
_item1">    <!--Second line item
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Item
"> etc.</F>    <!--Has the value "9122"
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Desc
ription"> etc.</F>    <!--Has the value "Widget"
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Cost
"> etc.</F>    <!--Has the value "8.25"
    </F>
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Line
_item2">    <!--Third line item (not a line item)
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Item
"> etc.</F>    <!--No data
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Desc
ription"> etc.</F>    <!--Has the value "Total"
    <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev226_Cost
"> etc.</F>    <!--Has the value "15.20"
    </F>
    etc.

```

Procedure

1. If the page you are processing has non-line item fields within the grid region, you might need to identify those non-line items and remove them. Datacap includes an action to identify and remove these items for you.

Library	Action	Description
Validations	CheckSubFields	Confirms that values exist in child fields of specified parent field. Deletes the parent field if any of the specified child fields have no values.

2. You must attach the rule to the line item grid object in the document hierarchy and ensure that the rule runs after recognition is complete. Use one of the following methods to attach the rule:
 - o Include the rule in the `Recognize` ruleset and attach it to the `Close` element of the line item grid object.
 - o Include the rule within a separate task profile that runs after recognition but before validation; for example, the `Clean` ruleset. Attach the rule to the `Open` element of the line item grid object.

When you use either method, Datacap recognizes that the `Item` field in the third line item has no value. Therefore, Datacap deletes the field, leaving only the two real line items.

Parent topic: [Handling line item grids](#)

Exporting data from a line item grid

You can export line item grid data by using the export actions. However, since the data exists at different levels in the runtime hierarchy, you need different rules for each level.

About this task

For example, to export the data from an invoice, you typically need:

- A rule to set up the export file or open the database
- A rule to export the header information
- A rule to export each of the line items
- A rule to export any trailing items (for example, the invoice total field)
- A rule to close the file or database

An example of exporting data to a database is provided in the topic [Exporting to a database](#).

For more information, see [Smart parameters](#).

Parent topic: [Handling line item grids](#)

TravelDocs: Adding new pages that contain line item grids

In this example, you update the TravelDocs application to process the Meals and Other Charges pages of the Hotel document type. Both of these pages include line item grids of undefined length.

- [Updating the document hierarchy](#)
When you set up the document hierarchy earlier, you skipped the Meals and Other_Charges pages. The business requirements specify the rules for the structure of the new page types.
- [Attaching the existing page rules to the new pages](#)
You need to add the same rules you used on the other pages in the TravelDocs application to the new pages.
- [Creating the page fingerprints](#)
The next step is to add fingerprints for the new page types to the fingerprint library.
- [Defining the recognition zones](#)
Next, you need to define the recognition zones for each of the new page types.

Parent topic: [Handling line item grids](#)

Updating the document hierarchy

When you set up the document hierarchy earlier, you skipped the Meals and Other_Charges pages. The business requirements specify the rules for the structure of the new page types.

About this task

The following rules are specified for the structure of the new page types:

	Number	Required?	Order
Hotel			

	Number	Required?	Order
Meals	Any number per document	No	Any position in the document
Other_Charges	Any number per document	No	Any position in the document

Within the document hierarchy, the following variables define the structure of the pages within the document.

	Maximum	Minimum	Order
Hotel			
Meals	0	0	0
Other_Charges	0	0	0

- [Adding pages to the document hierarchy](#)
You need to add new pages that contain the line item grids to the DCO.
- [Creating data fields](#)
The business requirements specification defines these fields for each new page type.

Parent topic: [TravelDocs: Adding new pages that contain line item grids](#)

Adding pages to the document hierarchy

You need to add new pages that contain the line item grids to the DCO.

Procedure

To add pages to the document hierarchy:

1. In the Document Hierarchy pane, click Lock DCO for editing to lock the document hierarchy for editing.
2. Expand the tree so you can see the document types.
3. Right-click the Hotel document node and choose Add multiple > Pages. Then, type 2 in the box and press Enter.
4. Rename the new pages from Page 1 and Page 2 to Meals and Other_Charges.
5. Click Save.
6. Right-click the Meals page node and choose Manage variables.
7. Make sure the Max, Min, and Order values are as specified in the preceding table (the Meals page is 0, 0, 0). Then, click Done.
8. Right-click on the Other_Charges page node and choose Manage variables. Enter the Max, Min, and Order values as specified in the preceding table (the Other_Charges page is also 0, 0, 0) and click Done.
9. Click Save.

Parent topic: [Updating the document hierarchy](#)

Creating data fields

The business requirements specification defines these fields for each new page type.

About this task

Meals	Other Charges
--------------	----------------------

Meals	Other Charges
Meals_Grid <ul style="list-style-type: none"> • Meals_Line_Item <ul style="list-style-type: none"> ◦ Date ◦ Description ◦ Cost Meals_Total	Other_Charges_Grid <ul style="list-style-type: none"> • Other_Charges_Line_Item <ul style="list-style-type: none"> ◦ Date ◦ Category ◦ Quantity ◦ Unit_Cost ◦ Total Other_Charges_Total

Procedure

1. Confirm that the document hierarchy is still locked for editing.
2. Right-click the Meals page and choose Add multiple > Fields. Then, type 2 in the box and press Enter on your keyboard.
3. Rename the new fields Meals_Grid and Meals_Total.
4. Right-click the Meals_Grid field and choose Add > Field.
5. Rename the new field Meals_Line_Item.
6. Right-click the Meals_Line_Item field and choose Add multiple > Fields. Then, type 3 in the box and press Enter.
7. Rename the new fields Date, Description, and Cost.
8. Repeat to add the fields and subfields for the Other_Charges page, as shown in the table. When you add the Date field, click Yes to inherit all rules and properties.
9. Click Save. The following example shows the document hierarchy for the new Hotel pages.

Meals

Open

- Meals_Grid
- Open
 - Meals_Line_Item
 - Open
 - Date
 - Description
 - Cost
 - Close
- Close
- Meals_Total

Close

Other_Charges

Open

- Other_Charges_Grid
- Open
 - Other_Charges_Line_Item
 - Open
 - Date
 - Category
 - Quantity
 - Unit_Cost
 - Total

- Close
- Close
- Other_Charges_Total

Close

Parent topic: [Updating the document hierarchy](#)

Attaching the existing page rules to the new pages

You need to add the same rules you used on the other pages in the TravelDocs application to the new pages.

Procedure

To attach the existing page rules to the new pages:

1. Make sure that the document hierarchy is still locked for editing.
2. Expand the document hierarchy so the Meals page and Other_Charges page are visible and then select the Meals page.
3. In the Rulesets pane, expand the CreateDocs ruleset, select the Create Fields rule, and click Add to DCO. Datacap adds the rule to the Open element of the Meals page.
4. Select the Other_Charges page and click Add to DCO. Datacap adds the rule to the Open element of the Other_Charges page.
5. Repeat to add the following rules to each page: Recognize: Recognize Page, Validate: Validate Page, Routing: Routing Rule 1, and Export: Export Page. Each page now has an Open element.
6. Click Save and then click Unlock DCO.

Parent topic: [TravelDocs: Adding new pages that contain line item grids](#)

Creating the page fingerprints

The next step is to add fingerprints for the new page types to the fingerprint library.

Procedure

To create the page fingerprints:

1. Click the Datacap Studio Zones tab.
2. In the Fingerprints pane, right-click the Hotel class and choose Add fingerprint.
3. Browse to the folder where the TravelDocs fingerprint images are located.
4. Select Hotel4.tif, and click Open. When asked if you want to enhance the image, click Yes.
5. In the Image Processing window, click Run image processing to apply the application's image-processing properties. Make sure that the lines disappear from the processed page.
6. Click Save, choose Save image, and click OK. Then, click x to close the Image Processing window.
7. With the new fingerprint selected, click the Type field and choose Meals.
8. Repeat to add Hotel5.tif and enhance the page image in the same way, but set the type to Other_Charges.

Parent topic: [TravelDocs: Adding new pages that contain line item grids](#)

Defining the recognition zones

Next, you need to define the recognition zones for each of the new page types.

Procedure

1. Define the zones on the Meals page.
 - a. In the Fingerprints pane, select the Meals page.
 - b. In the Image View pane, click Zoom to enlarge the page so you can see the fields clearly.
 - c. In the Document hierarchy pane, click Lock DCO for editing.
 - d. In the Document hierarchy pane, expand the Meals page so you can see all of the fields and subfields.
 - e. Select the Meals_Total field and draw a bounding box around the total cost beneath the line item grid.

Tip: If you create the Meals_Details zone first, you cannot draw the Meals_Total zone inside it. Therefore, you are drawing the Meals_Total zone first.
 - f. Select the Meals_Grid field. Then, draw a bounding box around the grid items. Because you do not know how many line items there might be on the actual page images, extend the bounding box to the bottom of the page.
 - g. Select the Meals_Line_Item field and draw a bounding box around the first line item.
 - h. Select the Meals_Line_Item >Date field and draw a box around the date in the first line item.
 - i. Repeat to create zones for the Description and Cost fields.
 - j. In the Document hierarchy pane, click Save.
2. Define the zones on the Other_Charges page.
 - a. In the Fingerprints pane, select the Other_Charges page.
 - b. Make sure that the document hierarchy is still locked for editing. Then, expand the Other_Charges page so that you can see all of the fields and subfields.
 - c. Select the Other_Charges_Total field and draw a bounding box around the total cost beneath the line item grid.
 - d. Select the Other_Charges_Grid field and draw a bounding box around the grid items. Extend the bounding box to the bottom of the page as you did for the Meals page.
 - e. Select the Other_Charges_Line_Item field and draw a bounding box around the first line item.
 - f. Select each of the subfields in turn and draw a bounding box around each field in the first line item.
 - g. In the Document hierarchy pane, click Save and then click Unlock DCO.

Parent topic: [TravelDocs: Adding new pages that contain line item grids](#)

TravelDocs: Recognizing line item grid data

Recognizing the data with a line item grid requires these rules. Each rule is attached to a different object in the document hierarchy.

About this task

- A rule that is attached to the line item grid object to scan all line items
- A rule that is attached to the line item object to scan each line item
- A rule that is attached to each field within the line item to read each field
- A rule that is attached to the grid total field to locate and read the total cost
- [Creating the recognition rules for the line items](#)

You need to create new recognition rules for the line items in the grid by using Rulemanager.
- [Creating the recognition rule for the grid total](#)

You need to create a recognition rule for the total amount that is listed as the sum of all the line item totals. You can then compare this total with the sum you get for the line item totals.

- [Attaching the rules to the document hierarchy](#)
After you create the rules, you will need to add them to the appropriate fields in the document hierarchy.
- [Running a batch through the workflow](#)
You can run a batch through the workflow to test the document hierarchy and the recognition rules you created.
- [Creating rules to remove the non-line items](#)
You need to create rules that remove non-line item values such as dates and descriptions.

Parent topic: [Handling line item grids](#)

Creating the recognition rules for the line items

You need to create new recognition rules for the line items in the grid by using Rulemanager.

Procedure

To create the recognition rules for the line items:

1. Click the Datacap Studio Rulemanager tab.
2. In the Rulesets pane, select the Recognize ruleset and click Lock/Unlock ruleset.
3. Right-click the Recognize ruleset and choose Add Rule. Rename the new rule Recognize Line Item Grid.
4. Right-click the Recognize ruleset and choose Add Rule. Rename the new rule Recognize Line Item.
5. Right-click the Recognize ruleset and choose Add Rule. Rename the new rule Recognize Line Item Field.
6. Under the Recognize Line Item Grid rule, select Function 1. Then, click the Actions library tab, expand the Zones library, select the ScanDetails action, and click Add to function.
7. Under the Recognize Line Item rule, select Function 1. Then, select the Scan LineItem action and click Add to function.
8. Under the Recognize Line Item Field rule, select Function 1. Then, select the PopulateZNLineItemField action and click the Add to function.
9. Click Save.

Parent topic: [TravelDocs: Recognizing line item grid data](#)

Creating the recognition rule for the grid total

You need to create a recognition rule for the total amount that is listed as the sum of all the line item totals. You can then compare this total with the sum you get for the line item totals.

Procedure

To create the recognition rule for the grid total:

1. Right-click the Recognize ruleset and choose Add Rule. Rename the new rule Recognize Grid Total Field.
2. Under the Recognize Line Item Grid rule, select Function 1.
3. On the Actions library tab, expand the Locate library, select the FindLastRegex action, and click Add to function.
4. In the Locate library, select the GoRightWord action and click Add to function.
5. In the Locate library, select the UpdateField action and click Add to function.
6. In the Rulesets pane, select the FindLastRegex action. Then, in the Properties pane, set strParam to Total.
7. In the Rulesets pane, select the GoRightWord action. Then, in the Properties pane, set strParam to 1.
8. Click Save and then click Lock/Unlock ruleset and choose Publish ruleset.

Attaching the rules to the document hierarchy

After you create the rules, you will need to add them to the appropriate fields in the document hierarchy.

About this task

To attach the rules to the document hierarchy:

- Attach the `Recognize Line Item Grid` rule to the `Meals_Grid` and `Other_Charges_Grid` fields.
- Attach the `Recognize Line Item` rule to the `Meals_Line_Item` and `Other_Charges_Line_Item` field.
- Attach the `Recognize Line Item Field` rule to each field.
- Attach the `Recognize Grid Total Field` rule to the `Meals_Total` and `Other_Charges_Total` fields.

Procedure

To attach the rules to the document hierarchy:

1. In the Document hierarchy pane, click Lock DCO for editing.
2. Expand the document hierarchy so you can see all of the fields and subfields on the `Meals` and `Other_Charges` pages.
3. Select the `Meals_Grid` field. Then, in the Rulesets pane, select the `Recognize Line Item Grid` rule and click Add to DCO. Datacap adds the rule to the Open element of the field.
4. Select the `Other_Charges_Grid` field and click Add to DCO.
5. Select the `Meals_Line_Item` field. Then, in the Rulesets pane, select the `Recognize Line Item` rule and click Add to DCO.
6. Select the `Other_Charges_Line_Item` field and click Add to DCO.
7. Select the `Date` field under the `Meals_Line_Item`. Then, in the Rulesets pane, select the `Recognize Line Item Field` rule and click Add to DCO.
8. Repeat to attach the `Recognize Line Item Field` rule to the `Description` and `Cost` fields under the `Meals_Line_Item` and the `Category`, `Quantity`, `Unit_Cost`, and `Total` fields under the `Other_Charges_Line_Item`.
9. Select the `Meals_Total` field. Then, in the Rulesets pane, select the `Recognize Grid Total Field` rule and click Add to DCO.
10. Select the `Other_Charges_Total` field and click Add to DCO.
11. In the Document hierarchy pane, click Save, then click Unlock DCO.

Parent topic: [TravelDocs: Recognizing line item grid data](#)

Running a batch through the workflow

You can run a batch through the workflow to test the document hierarchy and the recognition rules you created.

Procedure

1. Click the Datacap Studio Test tab.
2. In the Workflow pane, select the VScan task profile under Main Job.
3. Click New to start a new batch.

4. Click Process rules for target object and Advance to move the batch through to the Verify task (do not run the Verify task).
5. In the Runtime batch hierarchy pane, expand the last hotel document and confirm that pages TM000012 and TM000013 were identified correctly.
6. Expand the Meals page and then expand each of the line items to confirm that each line item contains data.

When you expand the last line item, you see a value for the `Cost` field only. This field is the `Total` field at the bottom of the grid. Because this field is not a real line item, you must create a rule to eliminate non-line items. For more information, see [Creating rules to remove the non-line items](#).

7. Expand the `Other_Charges` page and then expand each line item to confirm that each line item contains data. When you expand the last line item, you see a value for the `Total` field only. This field is the `Total` field at the bottom of the grid that must also be eliminated.
8. In the Workflow pane, right-click the batch and choose Cancel.

Parent topic: [TravelDocs: Recognizing line item grid data](#)

Creating rules to remove the non-line items

You need to create rules that remove non-line item values such as dates and descriptions.

Procedure

To create rules to remove the non-line items:

1. Click the Datacap Studio Rulemanager tab.
2. In the Rulesets pane, select the Clean ruleset and click Lock/Unlock ruleset.
3. Right-click the Clean ruleset and choose Add Rule. Rename the new rule `Remove Non Line Items (Meals)`.
4. Under the Remove Non Line Items (Meals) rule, select Function 1.
5. On the Actions library tab, expand the Validations library. Select the CheckSubFields action and then click Add to function.
6. In the Rulesets pane, select the CheckSubFields action. Then, in the Properties pane, set `strParam` as follows:

```
'Date' AND 'Description' AND 'Cost'
```
7. Right-click the Clean ruleset and choose Add Rule. Rename the new rule `Remove Non Line Items (Other Charges)`.
8. Under the Remove Non Line Items (Other Charges) rule, select Function 1.
9. Select the CheckSubFields action, and click Add to function.
10. In the Rulesets pane, select the new CheckSubFields action and set `strParam` as follows:

```
'Date' AND 'Category' AND 'Quantity' AND 'Unit_Cost' AND 'Total'
```
11. Click Save. Then, click Lock/Unlock ruleset and choose Publish ruleset.
12. In the Document hierarchy pane, click Lock DCO for editing.
13. Expand the Meals and `Other_Charges` pages.
14. Select the `Meals_Grid` field. Then, in the Rulesets pane, select the Remove Non Line Items (Meals) rule and click Add to DCO. Datacap adds the rule to the Open element of the field.
15. Select the `Other_Charges_Grid` field. Then, in the Rulesets pane, select the Remove Non Line Items (Other Charges) rule and click Add to DCO.
16. Click Save and then click Unlock DCO.

17. Run another batch through the workflow as described on the previous page and make sure that the non-line items are no longer included. Cancel the batch when you are done.

Parent topic: [TravelDocs: Recognizing line item grid data](#)

TravelDocs: Validating line item grid data

In the TravelDocs application, you add validations to the Other Charges page to confirm that the calculations on them are correct.

About this task

The grid on the Other Charges page includes calculated fields.

- The first `Total` field represents the Quantity that is multiplied by the Unit Cost.
- The second `Total` field represents the sum of all the line item totals.
- [Validating the line item totals](#)
Due to the way Datacap processes line item grids, you cannot attach calculations to line item objects. Instead, what you can do is create a field for calculation purposes.
- [Validating the grid total](#)
This validation ensures that the grid total (`Other_Charges_Total`) equals the sum of the line item totals (`Total`)
- [Running a batch through the workflow](#)
The Other Charges page includes calculation errors to trigger validation failures. When you run the batch through the workflow, you can see the validation failures.

Parent topic: [Handling line item grids](#)

Validating the line item totals

Due to the way Datacap processes line item grids, you cannot attach calculations to line item objects. Instead, what you can do is create a field for calculation purposes.

About this task

You used the `CalculateFields` action earlier when you checked that the total cost of the air ticket equals the airfare plus taxes.

Library	Action	Description
Validations	CalculateFields	Returns True if the arithmetic expression is valid; returns False otherwise.

Previously, you used the action to complete the calculation at the page level because the page was the next level up in the document hierarchy. In this case, the next level up in the hierarchy is the line item.

Because of the manner in which Datacap processes line item grids, you cannot attach calculations to line item objects. Instead, you can create a field for calculation purposes, for example, `Validation`, within the line item and attach the rule to this field.

- [Creating the validation rule](#)
You need to create a rule to check that the calculations on the Other charges page are correct.
- [Attaching the validation rule to the document hierarchy](#)
You need to create the field that you need to do validation and attach the new rule to it. In the document

hierarchy, this field is a subfield of the `Other_Charges_Line_Item` field.

Parent topic: [TravelDocs: Validating line item grid data](#)

Creating the validation rule

You need to create a rule to check that the calculations on the Other charges page are correct.

Procedure

1. On the Datacap Studio Rulemanager tab, in the Rulesets pane, select the Validate ruleset and click Lock/Unlock ruleset.
2. Right-click the Validate ruleset and choose Add Rule. Rename the new rule Validate Other Charge.
3. Under the Validate Other Charge rule, select Function 1.
4. On the Actions library tab, expand the Validations library. Select the CalculateFields action and click Add to function.
5. In the Rulesets pane, select the CalculateFields action. Then, in the Properties pane, set strParam as follows:

```
'Quantity' * 'Unit_Cost' = 'Total'
```

6. Click Save. Then, click Lock/Unlock ruleset and choose Publish ruleset.

Parent topic: [Validating the line item totals](#)

Attaching the validation rule to the document hierarchy

You need to create the field that you need to do validation and attach the new rule to it. In the document hierarchy, this field is a subfield of the `Other_Charges_Line_Item` field.

Procedure

1. In the Document hierarchy pane, click Lock DCO for editing.
2. Expand the Other_Charges page completely, so you can see all the individual fields.
3. Right-click the Other_Charges_Line_Item field and choose Add > Field. Then, rename the new field Validation.
4. Select the Validation field. Then, in the Rulesets pane, select the Validate Other Charge rule and click Add to DCO. Datacap adds the rule to the field's "Open" element.
5. Click Save.

Parent topic: [Validating the line item totals](#)

Validating the grid total

This validation ensures that the grid total (`Other_Charges_Total`) equals the sum of the line item totals (`Total`)

About this task

There are different methods to do this calculation. You can attach the following rule to the page's Close element:

The validation action looks a little unusual:

```
CalculateFields("'Total' = 'Other_Charges_Total'")
```

The reference to Total sums all of the child fields that are labeled Total. The action then compares the sum to the Other_Charges_Total field.

- [Creating the validation rule](#)
You can create a validation rule to ensure that the grid total (Other_Charges_Total) equals the sum of the line item totals (Total).
- [Attaching the rule to the document hierarchy](#)
You must attach the new validation rule to the Other_Charges page's Close element.

Parent topic: [TravelDocs: Validating line item grid data](#)

Creating the validation rule

You can create a validation rule to ensure that the grid total (Other_Charges_Total) equals the sum of the line item totals (Total).

Procedure

1. On the Datacap Studio Rulemanager tab, in the Rulesets pane, select the Validate ruleset and click Lock/Unlock ruleset.
2. Right-click the Validate ruleset and choose Add Rule. Rename the new rule `Validate Other Charges Total`.
3. Under the Validate Other Charges Total rule, select Function 1.
4. On the Actions library tab, expand the Validations library. Select the CalculateFields action and click Add to function.
5. In the Rulesets pane, select the CalculateFields action. Then, in the Properties pane, set strParam as follows:

```
'Total' = 'Other_Charges_Total'
```

6. Click Save, then click Lock/Unlock ruleset and choose Publish ruleset. Here is an example of the Validate Other Charges rule:
 - Validate Other Charges Total
 - Function1
 - CalculateFields("'Total' = 'Other_Charges_Total'")

Parent topic: [Validating the grid total](#)

Attaching the rule to the document hierarchy

You must attach the new validation rule to the Other_Charges page's Close element.

Procedure

1. Make sure that the document hierarchy is still locked for editing.
2. If necessary, expand the Other_Charges page so you can see the page's Close element.
3. Select the Close element. Then, in the Rulesets pane, select the Validate Other Charges Total rule and click Add to DCO. Datacap adds the rule to the Close element.
4. Click Save and then click Unlock DCO.

Parent topic: [Validating the grid total](#)

Running a batch through the workflow

The Other Charges page includes calculation errors to trigger validation failures. When you run the batch through the workflow, you can see the validation failures.

Procedure

1. Click the Datacap Studio Test tab.
2. In the Workflow pane, select the VScan task profile under Main Job.
3. Click New to start a new batch.
4. Click Process rules for target object and Advance to move the batch through to the Verify task (do not run the Verify task).
5. Open the application's most recent batch folder `C:\Datacap\TravelDocs\batches\batch_id`. Then, open the file `Ruleruner.xml` and scroll down to the data for page `TM000013`.

```
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev255
_TM000013">
<V n="TYPE">Other_Charges</V>
<V n="STATUS">1</V>          <-- Page status is '1'
etc.
<V n="MESSAGE">Failed Calculation:FormatNumber(138.75 ,8,0,0) <--Calc failure
FormatNumber( 238.75,8,0,0)</V>
<V n="DATAFILE">tm000013.xml</V>
</P>
```

The page status is '1' (indicating a problem) and the message indicates that the calculation for the `Total_Charges` field failed.

6. Open the file `tm000013.xml`. Notice that there is a calculation failure in the first line item and that Datacap flags all the fields that are involved in the calculation.

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev255
_Other_Charges_Line_Item0">
<V n="TYPE">Other_Charges_Line_Item</V>
etc.
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev255
_Quantity">
etc.
<V n="STATUS">1</V>
<V n="MESSAGE">Failed By Calculate Action On Field <-- Quantity field
&apos;Validation&apos;.</V>
etc.
</F>
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev255
_Unit_Cost">
etc.
<V n="STATUS">1</V>
<V n="MESSAGE">Failed By Calculate Action On Field <-- Unit_Cost field
&apos;Validation&apos;.</V>
etc.
</F>
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev255
_Total">
etc.
<V n="STATUS">1</V>
<V n="MESSAGE">Failed By Calculate Action On Field <-- Total field
```

```

&apos;Validation&apos;.</V>
etc.
</F>
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev255
_validation">
etc.
<V n="STATUS">1</V>
<V n="MESSAGE">Failed Calculation:FormatNumber <-- Validation field
(1 * 4.95 ,8,0,0)=FormatNumber( 9.9,8,0,0)</V>
</F>
</F>

```

7. In Datacap Studio, cancel the batch when you are done.

Parent topic: [TravelDocs: Validating line item grid data](#)

TravelDocs: Verifying the line item grid pages

For demonstration purposes, use Datacap Desktop to verify the line item grid pages.

- [Verifying pages by using Datacap Desktop](#)
To verify the line item grid pages, use Datacap Desktop, which defaults to the field-at-a-time interface.

Parent topic: [Handling line item grids](#)

Verifying pages by using Datacap Desktop

To verify the line item grid pages, use Datacap Desktop, which defaults to the field-at-a-time interface.

About this task

This procedure requires that you prepare a batch and place the batch on hold. For details, see [Preparing a batch for verification](#).

Procedure

1. In the Start menu, click IBM Datacap Clients and select Datacap Desktop.
2. In the Application field, type `TravelDocs`.
3. Enter `User ID: admin`, `Password: admin`, and `Station: 1` and click `Login`.
4. In the Shortcut field, select `Verify` and click `Start`. Then, open the most recent batch by double-clicking it.
5. In the Batch View pane, expand the third Hotel document and select the `Other_Charges` page. Then, select `Other_Charges_Line_Item0` in the grid. Datacap Desktop displays the corresponding line item in the snippet view beneath the grid.
6. Review the fields with validation errors that are marked in red.
7. Because the application is configured so that operators can override validation failures, click `Submit` and then click `OK` to override the errors.
Important: If you want to modify the Datacap Desktop interface, you must create a custom panel for each page. See *Datacap Desktop panel customization*.
8. When prompted to continue from the start of the batch, click `No` and then close Datacap Desktop.

Parent topic: [TravelDocs: Verifying the line item grid pages](#)

TravelDocs: Exporting line item grid data to a database

You can update the application to export line item grid data from the Other Charges page to a table in the export database.

- [Exporting to a database](#)

To export the line item grid data to a database, you need to create an export database table, add rules to the ExportDB ruleset, and attach the rules to the document hierarchy.

Parent topic: [Handling line item grids](#)

Exporting to a database

To export the line item grid data to a database, you need to create an export database table, add rules to the ExportDB ruleset, and attach the rules to the document hierarchy.

- [Creating the export database table](#)

You must use Microsoft Access to add the export database table to the TravelDocsExport.mdb file. You can skip this procedure if you previously used the sample Access file because the file that you copied includes the required table.

- [Adding rules to the ExportDB ruleset](#)

To successfully export line item grid data, you must add rules to the ExportDB ruleset and then add actions to the functions in these rules.

- [Attaching the Export Other rules to the document hierarchy](#)

After you add the rules to the ExportDB ruleset, you must attach them to the document hierarchy.

- [Running a batch through the workflow](#)

After you run a batch through the workflow, you can open the TravelDocsExport.mdb file and confirm that the line item grid data was exported to the `Other_Charges` table.

Parent topic: [TravelDocs: Exporting line item grid data to a database](#)

Creating the export database table

You must use Microsoft Access to add the export database table to the TravelDocsExport.mdb file. You can skip this procedure if you previously used the sample Access file because the file that you copied includes the required table.

Procedure

1. Open the file `C:\Datacap\TravelDocs\TravelDocsExport.mdb` in Microsoft Access.
2. Create a table that is called `Other_Charges`.
3. Create a field for `BatchID` and for each of the fields that are defined for the Other Charges page. Make all fields of type `Text`.
4. Save the new table.

Parent topic: [Exporting to a database](#)

Adding rules to the ExportDB ruleset

To successfully export line item grid data, you must add rules to the ExportDB ruleset and then add actions to the functions in these rules.

Procedure

1. In the Datacap Studio Rulesets pane, select the ExportDB ruleset and click Lock/Unlock ruleset for editing.
2. Right-click the ExportDB ruleset and choose Add Rule. Rename the new rule Export Other Open Database.
3. Repeat to create three more rules:
 - o Export Other Line Item
 - o Export Other Total
 - o Export Other Close Database
4. Click the Actions library tab and expand the ExportDB library.
5. Expand the Export Other Open Database rule and select Function1.
6. Select and add each of the actions to Function1 by clicking Add to function, and set the action parameters as shown in the table.

Action	Parameter
ExportOpenConnection	@APPVAR(*)/exportdb:cs)
SetTableName	Other_Charges

7. Expand the Export Other Line Item rule and select Function1.
8. Select and add each of the actions to Function1 by clicking Add to function, and then set the action parameters as shown in the table.

Action	Parameter
ExportBatchIDToColumn	BatchID
ExportFieldToColumn	Date,Charge_Date
ExportFieldToColumn	Category,Category
ExportFieldToColumn	Quantity,Quantity
ExportFieldToColumn	Unit_Cost,Unit_Cost
ExportFieldToColumn	Total>Total
AddRecord	

9. Expand the Export Other Total rule and select Function1.
10. Select and add each of the actions to Function1 by clicking Add to function, and set the action parameters as shown in the table.

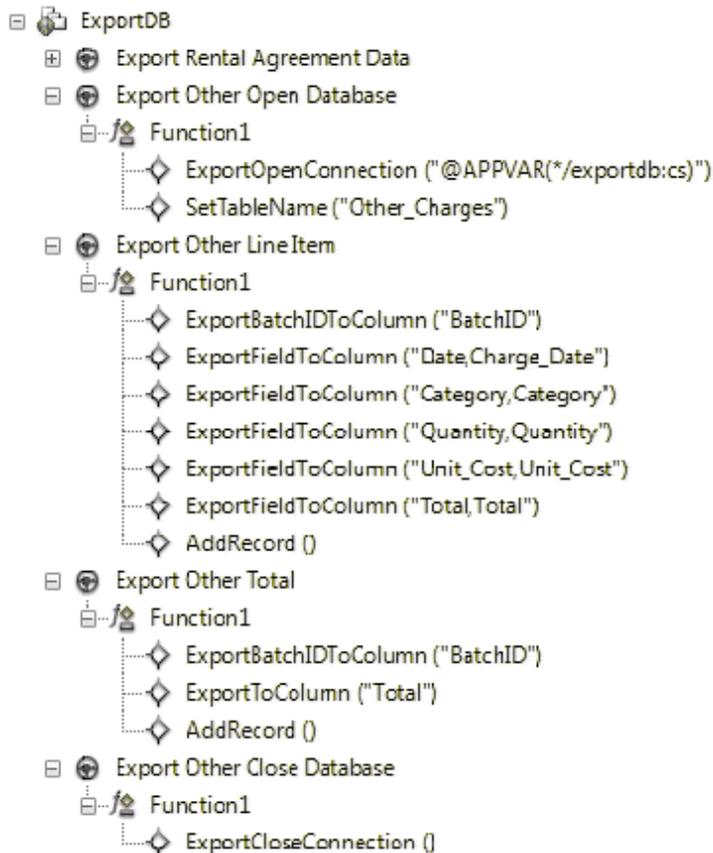
Important: The second action is ExportToColumn.

Action	Parameter
ExportBatchIDToColumn	BatchID
ExportToColumn	Total
AddRecord	

11. Expand the Export Other Close Database rule and select Function1.
12. Select and add the action to Function1 by clicking Add to function.

Action	Parameter
ExportCloseConnection	

13. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and choose Publish ruleset. The following image shows the finished ruleset:



Parent topic: [Exporting to a database](#)

Attaching the Export Other rules to the document hierarchy

After you add the rules to the ExportDB ruleset, you must attach them to the document hierarchy.

Procedure

1. In the Document hierarchy pane, click Lock DCO for editing.
2. Expand the document hierarchy so all of the fields in the `Other_Charges` page are visible.
3. Select the `Other_Charges` page. Then, select the `Export Other Open Database` rule and click the Add to DCO.
4. Select the `Other_Charges_Line_Item` field. Then, select the `Export Other Line Item` rule and click Add to DCO.
5. Select the `Other_Charges_Total` field. Then, select the `Export Other Total` rule and click Add to DCO.
6. Select the `Other_Charges` page's Close element. Then, select the `Export Other Close Database` rule and click Add to DCO.
7. In the Document hierarchy pane, click Save and then click Unlock DCO. The rules are linked to the document hierarchy.

Parent topic: [Exporting to a database](#)

Running a batch through the workflow

After you run a batch through the workflow, you can open the `TravelDocsExport.mdb` file and confirm that the line item grid data was exported to the `Other_Charges` table.

Procedure

1. Click the Datacap Studio Test tab.
2. In the Workflow pane, select the VScan task profile under Main Job.
3. Click New to start a new batch.
4. Click Process rules for target object and Advance to move the batch through the VScan, PageID, Profiler, Verify, and Export tasks.
5. Open the file C:\Datacap\TravelDocs\TravelDocsExport.mdb and review the exported data in the `Other_Charges` table.

Parent topic: [Exporting to a database](#)

Smart parameters

Smart parameters are action arguments that get evaluated at run time.

You used a few smart parameters already in the TravelDocs application. For example, when you created the ExportXML ruleset you used `@BatchID` to set the export file name equal to the ID of the current batch:

```
xml_SetFileName("@BatchID")
```

Since the value of `@BatchID` is different each time you run a batch through the workflow, you can create a unique export file for each batch.

Datacap provides various special variables (`@<variable_name>`) that you can use within smart parameters to access dynamic information at run time. You can use smart parameters to do the following tasks:

- Get information from the application configuration file
- Get (or set) the value of an object in the document hierarchy (typically a variable or a field value)
- Get job information such as the task name, ID of the operator who is running the batch
- Get system information such as the current date, time

Smart parameters can also include strings, navigation elements, and combinations of strings, navigation elements, and special variables.

You are going to look at smart parameters in more detail. You can update the TravelDocs application to export the line item grid data to an XML file. This update requires the use of various smart parameters.

- [General structure of a smart parameter](#)
A smart parameter can include any number of elements that are combined at run time to produce a single action argument.
- [Special variables to access application configuration settings](#)
The application configuration file, or .app file, stores the paths, connection strings, and other settings of the application. You can use special variables to access the application configuration settings.
- [Access to the runtime hierarchy](#)
You can access the runtime batch hierarchy indirectly through the Datacap Studio Test tab and directly by opening the runtime XML files.
- [Use other special variables](#)
You can use other special variables to access task information, job information, and other information.
- [TravelDocs: Exporting line item grid data to an XML file](#)
You can update the application to export data to an XML file by using smart parameters to access the runtime document hierarchy.

Parent topic: [Datacap application development](#)

Related information:

General structure of a smart parameter

A smart parameter can include any number of elements that are combined at run time to produce a single action argument.

Elements might include special variables, string constants, and navigation elements.

Restriction: Smart parameters do not work with all actions. Check the Action help in Datacap Studio for compatibility information.

Example 1

The following example shows an action with a single smart parameter argument that includes three smart parameter elements.

```
SetSourceDirectory("@Appath(vsca
nimagedir)+\+Input")
```

Table 1. Smart Parameter Elements in Example 1

Smart Parameter Element	Description
@APPPATH(vsca nimagedir)	Special variable that gets a setting from the application configuration (.app) file. See the Special variables to access application configuration settings topic.
\	String constant
Input	String constant

Elements are combined by using the '+' sign. At run time, Datacap first evaluates any special variables and then concatenates the elements to create a single string that becomes the action argument.

```
07:13:25.53 3 Smart Parameter elements found
07:13:25.53 Parsing Smart Parameter element {0} value: "@APPPATH(vsca
nimagedir)"
07:13:25.54 @APPPATH key root value: 'vsca
nimagedir'
07:13:25.54 @APPPATH looking for workflow key: '*/dco_TravelDocs/vsca
nimagedir'
07:13:25.54 workflow key found: 'C:\Datacap\TravelDocs\images'
07:13:25.54 Parsing Smart Parameter element {1} value: "\"
07:13:25.54 Parsing Smart Parameter element {2} value: "Input"
07:13:25.54 Smart Parameter return value: 'C:\Datacap\TravelDocs\images'
07:13:25.54 looking for:C:\Datacap\TravelDocs\images\Input
07:13:25.55 Action changes: Directory with source images:
C:\Datacap\TravelDocs\images\Input
```

It is incorrect to specify the argument as @APPPATH(vsca
nimagedir)+\Input. The '\ sign when followed by a string represents a navigation element. See the [Use navigation elements to access the runtime hierarchy](#) topic. In this example, you do not want to specify a navigation element. Instead, you want to concatenate the result of @APPPATH(vsca
nimagedir) with the string \Input by using +\+Input.

Example 2

The next example shows an action with two smart parameter arguments. Each argument includes one smart parameter element:

```
rrSet ("..\Pickup Location","@B.FieldValue")
```

Table 2. Smart Parameter Arguments in Example 2

Smart Parameter Argument	Description
..\Pickup_Location	Navigation element that references another field on the same page within the runtime hierarchy
@B.FieldValue	Special variable that references a batch level custom variable within the runtime hierarchy

The portion of the following log file shows how Datacap evaluates the first argument by recognizing it as a navigation element. It then goes to the referenced element within the runtime hierarchy and retrieves the value of the field. In this example, the action is bound to a field, so ..\Pickup_Location references another field at the same level on the same page.

```
08:17:30.892          action rrSet
08:17:30.892 (str="..\Pickup_Location",str="@B.FieldValue")
08:17:30.892          execute statement On Action Start
08:17:30.892          executing code:
08:17:30.892          Call OnActionStart()
08:17:30.892          /execute statement On Action Start
08:17:30.892 1 Smart Parameter element found
08:17:30.892 Parsing Smart Parameter element {0} value: "..\Pickup_Location"
08:17:30.892 DCO Parent Navigation key match (starts with '\' or '..\'). Calling
DCONavGetValue(..\Pickup_Location)
08:17:30.892 Finding Child 'Pickup_Location' -->
08:17:30.893 Found child 'Pickup_Location'
08:17:30.895 Finding Dictionary assigned to DCO Node:'Pickup_Location'
08:17:30.895 This DCO does not have an assigned Dictionary or is not an OMR type
Field.
08:17:30.895 Smart Parameter return value: 'Orlando (MCO)'
08:17:30.896 Setting '20110054.002.FieldValue' value to 'Orlando (MCO)'.
```

Parent topic: [Smart parameters](#)

Special variables to access application configuration settings

The application configuration file, or .app file, stores the paths, connection strings, and other settings of the application. You can use special variables to access the application configuration settings.

Do not attempt to modify this file directly, use the Datacap Application Manager. You used the Datacap Application Manager to configure the export database. For more information, see [Configuring the export database](#).

The .app file is stored in the root of the application folder. For example, the configuration file of the TravelDocs application is C:\Datacap\TravelDocs\TravelDocs.app:

```
<app name="TravelDocs" ver="45" modder="localadm.YODA647.DC14.DATACAP"
dt="03/09/12.753 11:41:06.753 " src_ver="1">
  <k name="tmservers">
    <k name="tms" ip="127.0.0.1" port="2402" retry="3"/>
  </k>
  <k name="runtime" v="batches"/>
  <k name="tmengine" cs="<encoded_connection_string>"/>          <-- encoded
  <k name="tmadmin" cs="<encoded_connection_string>"/>          <-- encoded
  <k name="dco_TravelDocs">
    <k name="setupdco" v="TravelDocs.xml"/>
    <k name="rules" v="rules"/>
    <k name="imagefix" v="imagefix.ini"/>
```

```

    <k name="UseFPXML" v="False"/>
    <k name="fingerprintconn" cs="<encoded_connection_string"/>      <-- encoded
    <k name="vsanimagedir" v="C:\Datacap\TravelDocs\images"/>        <-- encoded
    <k name="exportdb" cs="<encoded_connection_string"/>              <-- encoded
  </k>
  <k name="fingerprint" v="fingerprint"/>
  <k name="export" v="export"/>
</app>

```

Connection strings might contain user names and passwords, so they are encoded when they are written to the .app file. Datacap encodes and decodes user names and passwords automatically, so no special handling is required when you access them from your application by using smart parameters. For information about storing other action parameters in the .app file as encoded strings, see [Storing passwords, connection strings, and other parameters in the .app file](#).

Applications can access settings in the configuration file by using the following special variables:

Smart Parameter	Description
@APPPATH	Retrieves the path to a file or folder from the application configuration file.
@APPVAR	Retrieves a connection string, value, or other attribute from the application configuration file.

For detailed information about these and other special variables, see the [Smart Parameter Special Variable Reference](#).

For each special variable, you specify a key that represents the field that you want to get from the configuration file. When you used the APPPATH parameter, you specified export as the key (@APPPATH(export)).

- [Determining the correct key name](#)
You can obtain the correct special variable key name from the Datacap Application Manager.
- [Storing passwords, connection strings, and other parameters in the .app file](#)
The sample .app file illustrates how Datacap encodes the standard Datacap database connection strings (engine, admin, fingerprint, lookup, and export) before it writes them to the .app file.
- [Reference passwords, connection strings, and other parameters from your actions](#)
To reference the custom values from your actions, you must know the key path of the actions.

Parent topic: [Smart parameters](#)

Determining the correct key name

You can obtain the correct special variable key name from the Datacap Application Manager.

Procedure

To determining the correct key name:

1. Start the Datacap Application Manager. From the Windows Start menu, select IBM Datacap Services > Datacap > Datacap Datacap Application Manager.
2. Move the mouse pointer over the field. The smart parameter and key name are displayed in the tooltip.

The tooltip shows the path to the images folder of the application. The dco_* [1] prefix is required if the application has multiple workflows. Substitute * [1] with the workflow name, for example:

```
@APPPATH(dco_Workflow2/vsanimagedir)
```

If there is only one instance, you can use * instead, for example:

```
@APPPATH(* /vscaimagedir)
```

Keys for connection strings are more complicated because the connection string is stored in the `cs` attribute rather than the `v` attribute. The `v` attribute is the default attribute, so you do not need to specify the attribute name. To obtain the value of a different attribute, you must specify the attribute name by using `:<attribute_name>`.

You used the following syntax earlier to obtain the connection string for the lookup database of the application:

```
@APPVAR(* /lookupdb:cs)
```

Details of these special variables and a listing of key names are provided in the [Special variables to access application configuration settings](#) topic.

Parent topic: [Special variables to access application configuration settings](#)

Storing passwords, connection strings, and other parameters in the .app file

The sample .app file illustrates how Datacap encodes the standard Datacap database connection strings (engine, admin, fingerprint, lookup, and export) before it writes them to the .app file.

About this task

The function that is described in here is available in Datacap 8.0.1 or higher.

You can use the .app file to store other action parameters as encoded strings. You can then use smart parameters to access the strings from your actions. You do not have to specify sensitive information like passwords as action parameters.

- Instead of: `ex_login("svr/exch.asmx","user@company.com","secret")`
- Use: `ex_login("svr/exch.asmx","user@company.com",@APPVAR(values/adv/pwd))`

You can also use the .app file to store other action parameters that might not be sensitive. Action parameters that you do not want to hardcode into your actions. For example, you might choose to store a machine-specific path as a custom value. Then, you can change it easily if you move the application to a different computer.

Procedure

To store passwords, connection strings, and other parameters in the .app file:

1. In the Start menu, select IBM Datacap ServicesDatacap Application Manager.
2. Click the Custom values tab and select your application from the list on the left.
3. Click Add new value/CS name beneath the field you want to use:

Field	Description
General string values	Use this field for action parameters you do not want to hardcode in your actions. Instead of specifying a machine-specific path as an action parameter, enter the path here and reference it from your actions as described in the next section. Datacap encodes the values when it saves them to the .app file. Do not use this field for passwords as the strings are visible to anyone who is using the Datacap Application Manager. Use the Advanced values that are listed in this table.

Field	Description
Data source connection string values	Use this field to store data source connection strings that are not Datacap connection strings. Type or paste your connection string into this field and reference it from your actions as described in the next section.
Datacap data source connection string values	Use this field to store Datacap data source connection strings. Click the[...] to create the connection string by using Datacap supported providers and reference it from your actions as described in the next section.
Advanced values	Use this field to store passwords or other strings you do not want to reveal through the Datacap Application Manager. Values that you type here are masked. Reference the value from your actions as described in the next section.

4. Enter the value name and the value. Advanced values are masked whereas other values are not.

5. Close the Datacap Application Manager window.

Attention: If you change any of the settings in the application configuration file while Datacap Studio is open, click Connection Wizard to reopen your application. Then, you can run tasks from the Datacap Studio Test tab. Reconnecting to the application forces Datacap Studio to reload the information from the application configuration (.app) file.

Parent topic: [Special variables to access application configuration settings](#)

Related information:

[Application Manager](#)

Reference passwords, connection strings, and other parameters from your actions

To reference the custom values from your actions, you must know the key path of the actions.

You can get the key path from the help text on the Custom values tab in the Datacap Application Manager.

The text at the beginning of each section shows how to reference the value from an action. For example, for values that are defined in the Advanced values section, use `@APPVAR(values/adv/<value_name>)`. You can reference the value as:

```
@APPVAR(values/adv/MyPassword1)
```

The following table shows how to reference the values for each field type.

Field	Description
General string values	<code>@APPVAR(values/gen/<value_name></code> Example: <code>@APPVAR(values/gen/MyParameter1)</code>
Data source connection string values	<code>@APPVAR(values/dsn/<value_name>:cs)</code> Example: <code>@APPVAR(values/dsn/MyDatabase1:cs)</code>

Field	Description
Datacap data source connection string values	@APPVAR(values/tmdsn/<value_name>:cs) Example: @APPVAR(values/tmdsn/MyTMDatabase1:cs)
Advanced values	@APPVAR(values/adv/<value_name> Example: @APPVAR(values/adv/MyPassword1)

Attention: The :cs suffix is required to access connection strings that are defined in the Data source connection string and Datacap data source connection string fields.

Parent topic: [Special variables to access application configuration settings](#)

Access to the runtime hierarchy

You can access the runtime batch hierarchy indirectly through the Datacap Studio Test tab and directly by opening the runtime XML files.

The runtime XML file maps to the Runtime batch hierarchy in the Datacap Studio Test tab. For example the XML elements B id=, D id=, and P id=, all map to the Batch , Document, and Page in the batch hierarchy.

You access the information in the runtime hierarchy by using smart parameters.

- [Examples of using special variables to access the runtime hierarchy](#)
The TravelDocs application used special variables to access data in the runtime hierarchy. You can also use the ExportXML ruleset.
- [Summary of special variables for accessing the runtime hierarchy](#)
You can use @ID and @B/ @D/ @P<variable_name> special variables to get most of the batch, document, and page information from the runtime batch file.
- [Use navigation elements to access the runtime hierarchy](#)
In addition to using special variables, you can also use navigation elements to reference fields values and variables.

Parent topic: [Smart parameters](#)

Examples of using special variables to access the runtime hierarchy

The TravelDocs application used special variables to access data in the runtime hierarchy. You can also use the ExportXML ruleset.

The @BatchID example and the @ID example describe how to access data in the following sample runtime batch hierarchy XML from the ExportXML ruleset or with special variables.

```
<B
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev135_2011
0003.001">
  <V n="TYPE">TravelDocs</V>
  <D
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev135_2011
003.001.01">
  <V m="TYPE">Car_Rental</V>
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev135_TM00
0001">
  etc.
```

Use @BatchID to get the current batch ID

```
xml_SetFileName("@BatchID") returns xml_SetFileName("20110003.001")
```

Use @ID to get the ID of the current page

```
xml_NewNode("@ID,Rental_Agreements") returns xml_NewNode("@TM000001,Rental_Agreements")
```

Use @P*<field_name>* to get the value of a field on the current page

The @P variable retrieves the value of a field on a current page, as shown in this sample runtime page data XML file.

```
<P id=TM000001>
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev135_Pickup_Date">
  <V n="TYPE">Pickup_Date</V>
  <V n="Position">179,384,543,462</V>
  <V n="STATUS">0</V>
  <C cn="7" cr="200,416,220,430">84</C>      <!-- T -->
  <C cn="10" cr="226,425,240,440">117</C>    <!-- u -->
  <C cn="10" cr="245,425,258,440">101</C>    <!-- e -->
  <C cn="10" cr="260,425,270,440">115</C>    <!-- s -->
  <C cn="10" cr="273,435,278,444">44</C>    <!-- , -->
  <C cn="10" cr="336,419,337,440">32</C>    <!--   -->
  <C cn="10" cr="290,419,306,440">68</C>    <!-- D -->
  <C cn="10" cr="310,425,324,440">101</C>   <!-- e -->
  <C cn="10" cr="325,425,336,440">99</C>    <!-- c -->
  <C cn="10" cr="370,419,371,444">32</C>    <!--   -->
  <C cn="10" cr="349,419,363,440">55</C>    <!-- 7 -->
  <C cn="10" cr="365,435,370,444">44</C>    <!-- , -->
  <C cn="10" cr="445,419,446,440">32</C>    <!--   -->
  <C cn="10" cr="381,419,395,440">50</C>    <!-- 2 -->
  <C cn="10" cr="396,419,411,440">48</C>    <!-- 0 -->
  <C cn="10" cr="415,419,428,440">49</C>    <!-- 1 -->
  <C cn="10" cr="430,419,445,440">48</C>    <!-- 0 -->
```

Using the sample XML, `xml_SetModeValue("Pickup_Date,@P\Pickup_Date")` results in `xml_SetModeValue("Pickup_Date,Tues, Dec 7, 2010")`.

Parent topic: [Access to the runtime hierarchy](#)

Summary of special variables for accessing the runtime hierarchy

You can use @ID and @B/ @D/ @P<variable_name> special variables to get most of the batch, document, and page information from the runtime batch file.

For a full listing of special variables for accessing the runtime hierarchy, see [Special variables for accessing the runtime hierarchy](#). The following examples illustrate how to use special variables to access the runtime hierarchy.

Table 1. Special variables for accessing the runtime hierarchy

Access this part of the runtime hierarchy...	...by using this special variable	Hierarchy level
<?xml-stylesheet type="text/xsl" href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_...\.dc\o.xsl"?>	Not applicable	BATCH
<B id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev281_appdevguide_20110003.001">	@ID	BATCH
<V n="TYPE">TravelDocs</V>	@B.TYPE	BATCH
<V n="LAST_RR_TPROFILE">Rulerunner:m:eRun</V>	@B.LAST_RR_PROFILE	BATCH
<V n="STATUS">1</V>	@B.STATUS	BATCH
<D id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev281_appdevguide_20110003.001.01">	@ID	DOCUMENT
<V n="TYPE">Car_Rental</V>	@D.TYPE	DOCUMENT
<V n="STATUS">0</V>	@D.STATUS	DOCUMENT
<P id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev281_appdevguide_TM000001">	@ID	PAGE
<V n="TYPE">Rental_Agreement</V>	@P.TYPE	PAGE
<V n="STATUS">1</V>	@P.STATUS	PAGE
<V n="IMAGEFILE">tm000001.tif</V>	@P.IMAGEFILE	PAGE
<V n="ScanSrcPath">c:\...\images\page_01.tif</V>	@P.ScanSrcPath	PAGE
<V n="RecogStatus">0</V>	@P.RecogStatus	PAGE
<V n="Confidence">0.9660463</V>	@P.Confidence	PAGE
<V n="TemplateID">556</V>	@P.TemplateID	PAGE
<V n="DATAFILE">tm000001.xml</V>	@P.DATAFILE	PAGE
</P>	Not applicable	PAGE

Similarly, you can get most of the field information from the runtime page data file by using the @ID, @F, <variable_name>, and @P\<field_name> special variables, for example.

Table 2. Special variables for accessing field data in the runtime hierarchy

Access this part of the runtime hierarchy...	...by using this special variable	Hierarchy level
<F id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev281_appdevguide_Pickup_Date">	@ID	FIELD
<V n="TYPE">Pickup_Date</V>	@F.TYPE	FIELD
<V n="Position">179,394,543,462</V>	@F.Position	FIELD
<V n="STATUS">0</V>	@F.STATUS	FIELD

Access this part of the runtime hierarchy...	...by using this special variable	Hierarchy level
<C cn="7" cr="200,416,220,440">84</C>	@P\Pickup_Date	FIELD
<C cn="10" cr="226,425,240,440">117</C>	@P\Pickup_Date	FIELD
<C cn="10" cr="245,425,258,440">101</C>	@P\Pickup_Date	FIELD
<C cn="10" cr="260,425,270,440">115</C>	@P\Pickup_Date	FIELD
</F>	Not applicable	FIELD

Parent topic: [Access to the runtime hierarchy](#)

Use navigation elements to access the runtime hierarchy

In addition to using special variables, you can also use navigation elements to reference fields values and variables.

Smart parameters support the following navigation elements:

Table 1. Smart parameter navigation elements

Example	Description
\< field_name >	References a field that is 1 level beneath the current object
..\< field_name >	References a field at the same level as the current object

When you reference a field by specifying just the name of the field, Datacap retrieves the text value of the field. You can obtain the value of a variable that is associated with the field by appending `<variable_name>`, for example: `..\Car_Type.TYPE`.

You cannot use this syntax to access variables on parent objects.

Examples

In these examples, the `rr_Get` action is bound to a field.

- In the first example, the smart parameter returns the text of the `Car_Type` field on the current page.
- In the second example, the smart parameter returns the value of the `TYPE` variable of the `Car_Type` field.

```
Action: rr_Get("../Car_Type")      <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev371_Car_Type">
Return value: SUV                  <V n="TYPE">Car_Type</V>
<C cn="10" cr="588,748,600,769">83</C>  <-- ASCII 'S'
Action: rr_Get("../Car_Type.TYPE")
<C cn="10" cr="605,748,620,769">85</C>  <-- ASCII 'U'
Return value: Car_Type
<C cn="10" cr="625,748,643,769">86</C>  <-- ASCII 'V'
</F>
```

Parent topic: [Access to the runtime hierarchy](#)

Use other special variables

You can use other special variables to access task information, job information, and other information.

- [Access job and task information](#)
Datacap includes several special variables for accessing job and task information.
- [Access other information](#)
You can use special variables to access other information such as date, time, DCO and Pilot objects information, and application settings.

Parent topic: [Smart parameters](#)

Access job and task information

Datacap includes several special variables for accessing job and task information.

The special variables can be used to access the following task and job information.

- The ID and name of the current job
- The ID and name of the current task
- The user name and station that is associated with the job

A listing of these special variables is provided in [Special variables for accessing job and task information](#).

Parent topic: [Use other special variables](#)

Access other information

You can use special variables to access other information such as date, time, DCO and Pilot objects information, and application settings.

Datacap includes more special variables for accessing the following information.

- The current date and time
- Information from the DCO and Pilot objects
- Settings that are defined in the Paths.ini file of the application

A listing of these special variables is provided in [Miscellaneous special variables](#).

Parent topic: [Use other special variables](#)

TravelDocs: Exporting line item grid data to an XML file

You can update the application to export data to an XML file by using smart parameters to access the runtime document hierarchy.

You added functions to the TravelDocs application to export the Other Charge grid data to a database. For more information, see [Export to a database](#). You also set up a custom variable to store data within the document hierarchy. Now, you can export that same data to an XML file.

- [Adding rules to the ExportXML ruleset](#)
To export line item grid data to an XML file, you must add the appropriate rules to the ExportXML ruleset.
- [Attaching the Export Other XML rules to the document hierarchy](#)
After you add the appropriate rules to the ExportXML ruleset, you must attach the Export Other XML rules to the document hierarchy.
- [Running a batch through the workflow](#)

Adding rules to the ExportXML ruleset

To export line item grid data to an XML file, you must add the appropriate rules to the ExportXML ruleset.

Procedure

Adding rules to the ExportXML ruleset:

1. In the Datacap Studio Rulesets pane, select the ExportXML ruleset and click Lock/Unlock ruleset for editing.
2. Right-click the ExportXML ruleset and choose Add Rule. Rename the new rule `Export Other XML Page Node`.
3. Right-click the ExportXML ruleset and choose Add Rule. Rename the new rule `Export Other XML Line Item`.
4. Right-click the ExportXML ruleset and choose Add Rule. Rename the new rule `Export Other XML Total Cost`. The ExportXML ruleset includes the following rules:
 - o Open XML File
 - o Export Rental Agreement XML
 - o Export Other XML Page Node
 - o Export Other XML Line Item
 - o Export Other XML Total Cost
 - o Close XML File
5. Expand the Open XML File rule and the Open XML function.
6. Click the Actions library tab and expand the Export XML library.
7. Select and add each of the following actions that are shown in the following table to the end of the Open XML function by clicking Add to function. Then, set the action parameters as shown in the second table.

Action	Parameter
xml_NewNode	Car_Rentals,BatchID_+@BatchID
xml_NewNode	Rental_Agreements,Car_Rentals
xml_NewNode	Flights,BatchID_+@BatchID
xml_NewNode	Hotels,BatchID_+@BatchID
xml_NewNode	Other_Charges,Hotels

8. Expand the Export Other XML Page Node rule and select Function1.
9. Select and add each of the actions that are shown in the following table to Function1 by clicking Add to function. Then, set the action parameters as shown in the table.

Library	Action	Parameter
ExportXML	xml_NewNode	@ID,Other_Charges
rrunner	rrSet (do not use rr_Set)	varSource = @ID varTarget = @P.ID

Important: `xml_NewNode("@ID,Other_Charges")` creates a new XML node by using the ID of the current page (for example, `<TM000013>`) beneath the `<Other_Charges>` node.

`rrset("@ID","@P.ID")` stores the ID of the current page in a variable that is called *ID* within the runtime hierarchy (for example, `<V n="ID">TM000013</V>`).

10. Expand the Export Other XML Line Item rule and select Function1.

11. Select and add each of the actions that are shown in the following table to Function1 by clicking Add to function. Then, set the action parameters as shown in the table.

Library	Action	Parameter
ExportXML	xml_NewNode	Item,@P.ID
ExportXML	xml_SetAttributeValue	Item,Category,@F\Category
ExportXML	xml_SetAttributeValue	Item,Cost,@F\Total

Important: `xml_NewNode("Item,@P.ID")` creates a new node `<Item>` that is a child of the node that you created in the Export Other XML Page Node rule. The `@P.ID` parameter identifies the parent node by using the `ID` variable of the page that you saved earlier by using `rrset`. You cannot reference the ID of a parent object by using a smart parameter.

`xml_SetAttributeValue("Item,Category,@F\Category")` creates a `Category` attribute on the current `<Item>` node and sets the value to the value of the current line item's `Category` field (for example, `<Item Category=Internet/>`).

The `xml_SetAttributeValue("Item,Cost,@F\Total")` parameter is the same except that the attribute is `Cost` and the value is the current line item's `Total` field (for example, `<Item Cost="9.90"/>`).

12. Expand the Export Other XML Total rule and select Function1.
13. Select and add each of the actions in the following table to Function1 by clicking Add to function. Then, set the action parameters as shown in the table.

Library	Action	Parameter
ExportXML	xml_NewNode	Total_Cost,@P.ID
ExportXML	xml_SetNodeValue	Total_Cost,@P\Other_Charges_Total

Important: `xml_NewNode("Total_Cost,@P.ID")` creates a new XML node that is called `<Total__Cost>` beneath the page node that you created in the Export Other XML Page Node rule. The `xml_SetNodeValue` action sets the value of this node to the value of the current page's `Other_Charges_Total` field (for example, `<Total_Cost>$238.75</Total_Cost>`).

14. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and choose Publish ruleset.

Parent topic: [TravelDocs: Exporting line item grid data to an XML file](#)

Attaching the Export Other XML rules to the document hierarchy

After you add the appropriate rules to the ExportXML ruleset, you must attach the Export Other XML rules to the document hierarchy.

Procedure

To attaching the Export Other XML rules to the document hierarchy:

1. In the Document hierarchy pane, click Lock DCO for editing.
2. Expand the document hierarchy so that the fields in the Other_Charges page are visible.
3. Select the Other_Charges page. Then, select the Export Other XML Page Node rule and click Add to DCO.
4. Select the Other_Charges_Line_Item field. Then, select the Export Other XML Line Item rule and click Add to DCO.

5. Select the Other_Charges_Line_Total field. Then, select the Export Other XML Total Cost rule and click Add to DCO.
6. In the Document hierarchy pane, click Save. Then, click Unlock DCO. The rules are linked to the document hierarchy.

Parent topic: [TravelDocs: Exporting line item grid data to an XML file](#)

Running a batch through the workflow

Procedure

To run a batch through the workflow:

1. Click the Datacap Studio Test tab.
2. In the Workflow pane, select the VScan task profile under Main Job.
3. Click New to start a new batch.
4. Click Process rules for target object.
5. Click Advance to move the batch through the entire workflow.
6. Open the file C:\Datacap\TravelDocs\export*<batch_id>*.xml and review the exported XML data.

```
<?xml version='1.0' ?>
<BatchID_20100351.006>
  <Flights/>
  <Car_Rentals>
    <Rental_Agreements>
      <TM000001>
        <Pickup_Date>Trues, Dec 7, 2010</Pickup_Date>
        etc.
      </TM000001>
      <TM000003>
        <Pickup_Date>Mon, Dec 6, 2010</Pickup_Date>
        etc.
      </TM000003>
      <TM000004>
        <Pickup_Date>Mon, Dec 13, 2010</Pickup_Date>
        etc.
      </TM000004>
    </Rental_Agreements>
  </Car_Rentals>
  <Hotels>
    <Other_Charges>
      <TM000013>
        <Item Category="Internet" Cost="$9.90"/>
        <Item Category="Laundry" Cost="$18.00"/>
        <Item Category="Internet" Cost="$4.95"/>
        <Item Category="Newspaper" Cost="$2.00"/>
        <Item Category="Mini bar" Cost="$8.00"/>
        <Item Category="Internet" Cost="$4.95"/>
        <Item Category="Newspaper" Cost="$2.00"/>
        <Item Category="Mini bar" Cost="$8.00"/>
        <Item Category="Internet" Cost="$4.95"/>
        <Item Category="Newspaper" Cost="$2.00"/>
        <Item Category="Parking" Cost="$74.00"/>
        <Total_Cost>$238.75</Total_Cost>
      </TM000013>
    </Other_Charges>
  </Hotels>
</BatchID_20100351.006>
```

Parent topic: [TravelDocs: Exporting line item grid data to an XML file](#)

Text matching

You can add flexibility to your applications by using text matching to identify pages and locate data.

You used fingerprints to identify pages and recognition zones to locate data on those pages. The one exception was when you used the RegExFind action to locate the Total Cost field when you process line item grids. You saw in that example how you can use text matching to locate data that is not in a predictable location on the page.

To use text matching, you first do full page OCR on the incoming page. You can then search the recognition results for specific text. For example, if a page contains the words Car rental agreement, then the chances are high that it is a car rental agreement page. Similarly, if you locate a string Pickup date, then the chances are high that beside or below the string is the actual pickup date.

You can identify pages and locate data by using text matching. You can update the TravelDocs application to identify and process a new car rental agreement page by using text matching.

- [Identify pages with text matching](#)
Text matching uses the full page recognition results to identify pages.
- [Locate data with text matching](#)
You can locate data without using recognition zones. You search the full page recognition results for some static text that is next to the field you want to read.
- [TravelDocs: Update the application to use text matching](#)
You can update the TravelDocs application to use text matching for data recognition.

Parent topic: [Datacap application development](#)

Identify pages with text matching

Text matching uses the full page recognition results to identify pages.

You can identify pages by searching the recognition results for a string that is unique to each page type.

If all of the actions in a function return True, Datacap does not run any other functions in the current rule. For example, if you get a match on the word Car and can set the page type successfully. The PageID rule exits without running any of the other tests

Text matching uses the full page recognition results, so you must do full page OCR (or ICR) before you run any of the text matching actions. You can then use the WordFind action to determine whether a specific string is present and the SetPageType action to set the page type.

Library	Action	Description
Locate	WordFind	Locates the first (or next) occurrence of the specified word or phrase on the current page.
DCO	SetPageType	Assigns a page type to the current page in the runtime hierarchy.

The WordFind action is case-sensitive. Additionally, if different variants of a page type have different unique identifiers, you might need to use a more flexible matching technique. Such as regular expressions or keyword lists. For more information, see [Use regular expressions](#) and [Text matching with keyword lists](#).

Parent topic: [Text matching](#)

Related reference:

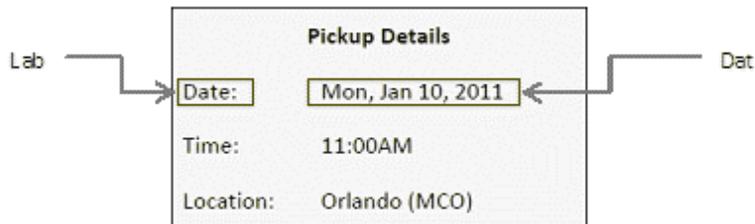
[Use regular expressions](#)

[Text matching with keyword lists](#)

Locate data with text matching

You can locate data without using recognition zones. You search the full page recognition results for some static text that is next to the field you want to read.

Forms typically have a label beside each field. When you locate the label, you can locate the adjacent data and then update the runtime hierarchy.



In this example, you locate the word Date. Then, you go to the right to obtain the matching data, and then write the information to the runtime hierarchy.

- [Locate simple strings](#)
The `Locate` library includes actions that you can use to locate specific text strings on the current page.
- [Use regular expressions](#)
If you use text matching on pages with various labels, you might be unable to locate a label by using a simple text string. In this case, you can use a regular expression.
- [Text matching with keyword lists](#)
If the page on which you are text matching has too many variations on the page, regular expressions might be unwieldy. A keyword list might be a better option.
- [Locate the field data](#)
After you locate the label, you must locate the adjacent field data.
- [Update the runtime data file with the recognized text](#)
After you locate the data field, you must write the data to the runtime hierarchy.
- [Text matching for data recognition limitations](#)
While text matching provides a quick way to read data from a non-fingerprinted page, it does have limitations.

Parent topic: [Text matching](#)

Locate simple strings

The `Locate` library includes actions that you can use to locate specific text strings on the current page.

The `Locate` library includes the following actions to locate specific text strings on a page.

Library	Action	Description
Locate	WordFind	Locates the first (or next) occurrence of the specified word or phrase on the current page.
Locate	FindLastWord	Locates the last occurrence of the specified word or phrase on the current page.

For more information on all of the actions in the `Locate` library, select the action on the Actions Library tab and click Display information.

The `WordFind` action is used in the previous page identification example. You can use `WordFind` or `FindLastWord` to locate any word or phrase on the current page. These actions require an exact match. If you need more flexible matching, you can use regular expressions or keyword lists.

Parent topic: [Locate data with text matching](#)

Use regular expressions

If you use text matching on pages with various labels, you might be unable to locate a label by using a simple text string. In this case, you can use a regular expression.

For example, one car rental company might use the label `Date`, another might use the label `Pickup date`, while another might use `Pickup Date`. In this example, you can use the `Locate` library regular expression actions, including the following actions.

Library	Action	Description
Locate	RegExFind	Same as <code>WordFind</code> , except that it supports regular expressions.
Locate	FindLastRegEx	Same as <code>FindLastWord</code> , except that it supports regular expressions.

One way you can search for any of the three pickup date labels in the previous example is by using a single regular expression.

```
RegExFind( "(Date) | (Pickup [Dd]ate) ")
```

The `Locate` library regular expression actions use the VBScript regular expression rules. Some of this usage is described in detail in the following MSDN article

<http://msdn.microsoft.com/en-us/library/1400241x%28v=vs.85%29.aspx>

Parent topic: [Locate data with text matching](#)

Text matching with keyword lists

If the page on which you are text matching has too many variations on the page, regular expressions might be unwieldy. A keyword list might be a better option.

Some labels might have more variability than you can address by using regular expressions. For example, if you are processing invoices from different vendors, one company might use the label `Total Cost`, another might use `Amount Due`, and another might use `Invoice total`.

The `Locate` library provides actions to do text matching by using either a keyword text file or a database.

Library	Action	Description
Locate	FindKeyList	Locates the first (or next) occurrence of a word or phrase that matches one of the entries in a keyword file.
Locate	FindLastKeyList	Locates the last occurrence of a word or phrase that matches one of the entries in a keyword file.
Locate	FindDBList	Locates a word that matches one of a list of words that are obtained from a SQL query.

In the invoice example, you can include all three labels and any others in a keyword text file. You can then use FindRegExList to locate the matching field. The keyword file must be a plain text file with the extension .key. The file must be in the application *dco_application_name* folder, unless you specify the full path to an alternative location. An example file might be named TotalCost.key and include variations on the theme such as:

- Total Cost
- Total cost
- TOTAL COST
- Amount Due
- Amount due
- AMOUNT DUE
- Total Amount
- Total amount
- TOTAL AMOUNT
- Total amount due
- Total Amount Due
- TOTAL AMOUNT DUE
- Invoice total
- Invoice Total
- INVOICE TOTAL

The locate library action is FindRegExList("TotalCost.key")

Tip: You can also include regular expressions in the keyword list.

Parent topic: [Locate data with text matching](#)

Locate the field data

After you locate the label, you must locate the adjacent field data.

The field data is usually to the right of the label, but it might also be above or below the label. Additionally, you might need to group words together if the data you are searching for includes spaces.

The full page recognition engine organizes the recognition results in the CCO file as a coordinate-based grid of lines and words. Each word is assigned a different position in the grid.

	1	2	3	4	5	6	7	8	9	10
1	Car	Rental	#4							
2	Pickup	Details	Return	Details						
3	Date:	Mon,	Jan	10,	2011	Date:	Fri,	Jan	14,	2011
4	Time:	11:00AM	Time:	04:00PM						
5	Location:	Orlando	(MCO)	Location:	Orlando	(MCO)				

This structure allows you to move around the recognition results by using the Locate library's navigation actions. The library also includes actions for grouping words together.

Library	Action	Description
Locate	GoRightWord	Moves the specified number of words to the right of the previously found word or phrase.

Library	Action	Description
Locate	GoDown(Up)Line	Moves down (up) the specified number of lines from the previously found word or phrase and selects the first word.
Locate	GroupWordsRIGHT(LEFT)	Groups words to the right (left) of the previously found word if they are no more than the specified number of character widths apart.
Locate	GroupWords	Groups words to the left and right of the previously found word if they are no more than the specified number of character widths apart.

The TravelDocs Recognize ruleset contains a rule that searches for the word Date and then goes one word to the right to obtain the data. The GroupWordsRIGHT action is required because, without it, you would get only the first word (Mon, in the Car Rental #4 example). The parameter 2 instructs the rule to group words that are two or fewer character widths apart.

Parent topic: [Locate data with text matching](#)

Update the runtime data file with the recognized text

After you locate the data field, you must write the data to the runtime hierarchy.

You write data to the runtime hierarchy by using the UpdateField action from the Locate library. The UpdateField action updates the page data file with the recognized value and position of the located word.

The CreateFields action is responsible for setting up each field in the runtime hierarchy. Initially both the field position and the field data are empty, as shown in the left column on the following example. The right column shows the field after it is populated by using the UpdateField action. The position information is used later to display the corresponding image snippet to the operator during verification.

After CreateFields():	After UpdateField():
------------------------------	-----------------------------

After CreateFields():	After UpdateField():
<pre> <F id="_dcs_markdown_workspace_Transform_htm lout_0_com.ibm.dc.develop.doc_dcdev348_Pi ckup_Date"> <V n="TYPE">Pickup_Date</V> <V n="Position">0,0,0,0</V> <V n="STATUS">0</V> </F> </pre>	<pre> <F id="_dcs_markdown_workspace_Transform_htm lout_0_com.ibm.dc.develop.doc_dcdev348_Pi ckup_Date"> <V n="TYPE">Pickup_Date</V> <V n="Position">539,419,789,452</V> <V n="STATUS">0</V> <C cn="10" cr="543,423,565,444">77</C> M <C cn="10" cr="570,429,585,444">111</C> o <C cn="10" cr="588,429,600,444">110</C> n <C cn="9" cr="605,440,610,448">44</C> , <C cn="9" cr="0,0,0,0">32</C> <C cn="9" cr="620,423,628,444">74</C> J <C cn="10" cr="630,429,643,444">97</C> a <C cn="10" cr="648,429,660,444">110</C> n <C cn="9" cr="0,0,0,0">32</C> <C cn="10" cr="674,423,685,444">49</C> 1 <C cn="10" cr="689,423,704,444">48</C> 0 <C cn="9" cr="705,440,710,448">44</C> , <C cn="9" cr="0,0,0,0">32</C> <C cn="10" cr="723,423,735,444">50</C> 2 <C cn="10" cr="739,423,754,444">48</C> 0 <C cn="10" cr="756,423,769,444">49</C> 1 <C cn="10" cr="774,423,785,444">49</C> 1 </F> </pre>

Parent topic: [Locate data with text matching](#)

Text matching for data recognition limitations

While text matching provides a quick way to read data from a non-fingerprinted page, it does have limitations.

The most notable limitations of text matching for data recognition are processing pages that include check box options or line item grids.

Parent topic: [Locate data with text matching](#)

TravelDocs: Update the application to use text matching

You can update the TravelDocs application to use text matching for data recognition.

To update the TravelDoc application to use text matching, you must determine which pages to process. Then, you can run data recognition and add the rules to the document hierarchy.

- [Identifying unrecognized pages by using text matching](#)
You can identify unrecognized pages by creating a ruleset with the `Identify using Text Match` function. You use the rule to recognize car rental pages by looking for text that is unique to that page type.
- [Recognizing data with text matching](#)
To recognize data by using text matching, add rules to the `Recognize` ruleset. These rules locate the data on the rental agreement pages that you identified by using text matching.
- [Attaching the rules to the document hierarchy](#)
You must attach each of the new `Recognize` rules to the corresponding field on the `Rental_Agreement` page definition.
- [Running a batch through the workflow](#)
Datacap provides an image file for a new rental agreement page that you can run through the workflow without fingerprinting.

Parent topic: [Text matching](#)

Identifying unrecognized pages by using text matching

You can identify unrecognized pages by creating a ruleset with the `Identify using Text Match` function. You use the rule to recognize car rental pages by looking for text that is unique to that page type.

Procedure

To identify unrecognized pages by using text matching:

1. In the Datacap Studio Rulesets pane, select the `PageID` ruleset and click `Lock/Unlock ruleset` for editing.
2. Expand the `PageID` rule.
3. Change the name of the existing function from `PageID: Other Function 1` to `Identify using Fingerprint`. Then, change the parameter on the `FindFingerprint` action to `False`.
Attention: Setting the parameter to `False` ensures that Datacap does not automatically generate a fingerprint file for unrecognized pages. If the current page does not match one of the existing fingerprints, this action fails and Datacap starts the next function, if there is one.
4. Right-click the `PageID` rule and choose `Add Function`. Then, rename the new function to `Identify using Text Match`.
5. Click the `Actions` library tab and add the actions that are shown in the following table to the `Identify using Text Match` function by clicking `Add to function`. Then, set the action parameters as shown.

Library	Action	Parameter
Locate	RegExFind	Car
Locate	RegExFind	Pickup
DCO	SetPageType	Rental_Agreement
rrunner	rrSet	varSource = Text varTarget = @P.MatchType

Important: The `Car` and `Pickup` parameters are tested to avoid mismatches on insurance pages. The `rrSet` action sets up a page variable that you must later identify which pages to process by using text matching.

Attention: If the `Identify using Fingerprint` function succeeds in identifying the page, the `Identify using Text Match` function does not start.

6. In the Rulesets pane, click `Save`. Then, click `Lock/Unlock ruleset` and choose `Publish Ruleset`.

Parent topic: [TravelDocs: Update the application to use text matching](#)

Recognizing data with text matching

To recognize data by using text matching, add rules to the Recognize ruleset. These rules locate the data on the rental agreement pages that you identified by using text matching.

About this task

Avoid running the rules on pages that were identified by using fingerprint matching. You can use the variable that you set up earlier to determine which pages use fingerprint matching.

Procedure

To recognize data by using text matching:

1. In the Datacap Studio Rulesets pane, select the Recognize rule set and click Lock/Unlock ruleset for editing.
2. Right-click the Recognize rule set and choose Add Rule. Repeat this step to add a total of six new rules.

Rename the rules as follows:

- Recognize Pickup Date
- Recognize Pickup Location
- Recognize Return Date
- Recognize Return Location
- Recognize Car Type
- Recognize Total Cost

3. Add the actions from the following table to Recognize Pickup Date > Function1 and set the action parameters as shown.

Library	Action	Parameter
rrunner	rrCompare	object1 = @P.MatchType object2 = Text
Locate	RegExFind	Date
Locate	GoRightWord	1
Locate	GroupWordsRIGHT	2
Locate	UpdateField	

4. Add the actions from the following table to Recognize Pickup Location > Function1 and set the parameters as shown.

Library	Action	Parameter
rrunner	rrCompare	object1 = @P.MatchType object2 = Text
Locate	RegExFind	Location
Locate	GoRightWord	1
Locate	GroupWordsRIGHT	2
Locate	UpdateField	

5. Add the actions from the following table to Recognize Return Date > Function1 and set the parameters as shown.

Library	Action	Parameter
rrunner	rrCompare	object1 = @P.MatchType object2 = Text
Locate	FindLastRegEx	Date
Locate	GoRightWord	1
Locate	GroupWordsRIGHT	2
Locate	UpdateField	

6. Add the actions from the following table to Recognize Return Location > Function1 and set the parameters as shown.

Library	Action	Parameter
rrunner	rrCompare	object1 = @P.MatchType object2 = Text
Locate	FindLastRegEx	Location
Locate	GoRightWord	1
Locate	GroupWordsRIGHT	2
Locate	UpdateField	

7. Add the actions from the following table to Recognize Car Type > Function1 and set the parameters as shown.

Library	Action	Parameter
rrunner	rrCompare	object1 = @P.MatchType object2 = Text
Locate	RegExFind	Car Type
Locate	GoRightWord	1
Locate	GroupWordsRIGHT	2
Locate	UpdateField	

8. Add the actions from the following table to Recognize Total Cost > Function1 and set the parameters as shown.

Library	Action	Parameter
rrunner	rrCompare	object1 = @P.MatchType object2 = Text
Locate	RegExFind	Total Cost
Locate	GoRightWord	1
Locate	UpdateField	

9. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and choose Publish Ruleset.
Attention: The use of text matching to locate and recognize OMR check box fields is beyond the scope of this activity. When you run the rule set, it leaves the OMR options in their default state ('0').

Parent topic: [TravelDocs: Update the application to use text matching](#)

Attaching the rules to the document hierarchy

You must attach each of the new Recognize rules to the corresponding field on the Rental_Agreement page definition.

Procedure

To attaching the rules to the document hierarchy:

1. In the Document hierarchy pane, click Lock DCO for editing.
2. Expand the document hierarchy so you can see the fields on the Rental_Agreement page.
3. Select the Pickup_Date field. In the Rulesets pane, select the Recognize Pickup Date rule and click Add to DCO.
4. Select the Pickup_Location field. In the Rulesets pane, select Recognize Pickup Location and the Add to DCO.
5. Select the Return_Date field. In the Rulesets pane, select Recognize Return Date and click Add to DCO.
6. Select the Return_Location field. In the Rulesets pane, select Recognize Return Location and click Add to DCO.
7. Select the Car_Type field. In the Rulesets pane, select Recognize Car Type and click Add to DCO.
8. Select the Total_Cost field. In the Rulesets pane, select Recognize Total Cost and click Add to DCO.
9. In the Document hierarchy pane, click Save, then click Unlock DCO.

Parent topic: [TravelDocs: Update the application to use text matching](#)

Running a batch through the workflow

Datacap provides an image file for a new rental agreement page that you can run through the workflow without fingerprinting.

Procedure

To run a batch through the workflow:

1. On the Datacap Studio Test tab, select the VScan task profile and click New.
2. Click Process rules for target object and click Advance to move the batch through the VScan and PageID task profiles.
3. On the Runtime batch hierarchy tab, select the first page (TM000001) and make sure the Car Rental #4 page is displayed on the Image tab. Confirm that the page type is Rental_Agreement.
4. Click Process rules for target object and click Advance to move the batch through the Rulerunner task profile.
5. On the Runtime batch hierarchy tab, expand the first Rental_Agreement page to confirm that the data was recognized correctly (except for the options).
6. In the Workflow pane, right-click the batch (it must be in the Verify task) and choose Hold.
7. Start the Datacap Web Client, select the TravelDocs application, and log on.
8. In the Datacap Web Client window, select the Monitor tab. Locate the most recent batch at the top of the list with a status of Hold.
9. Click the batch's row number at the left of the row and choose Yes to start the selected batch.
10. Review the data on the Car Rental #4 page. Then, quit the task and click OK to put the batch back on hold.
11. In Datacap Studio, in the Workflow pane, right-click the batch and choose Pick to change its status back to Running. Then, click Process rules for target object and click Advance to move the batch through the remainder of the workflow.

Pattern Matching

You can use Datacap pattern matching to identify pages and adjust misaligned or distorted images.

Standard fingerprint matching can compensate for minor page misalignment, but the offsets it can handle are small. If an incoming page is misaligned relative to the fingerprint, Datacap might not be able to identify the page. If Datacap does identify the page successfully, recognition might fail if the fields are not registered accurately. This issue is most problematic if the page contains OMR or hand print boxed data. But it can happen with any page, especially if the image is distorted during copying, scanning, or faxing.

Pattern matching actions use reference patterns, or anchor objects, that you define on the page fingerprints. The actions try to match those patterns to regions on the runtime pages. Pattern matching techniques that are available can use:

- Geometric patterns like page registration marks or vendor logos
- Text-based patterns

Pattern matching can be used to update the TravelDocs application to handle misaligned pages.

- [Pattern matching overview](#)
Pattern matching uses anchor objects that you define on the page fingerprints. These anchor objects can be geometric patterns, like a page registration marks or vendor logos, or text-based patterns.
- [Anchor objects setup](#)
An anchor object is a region of a fingerprint image that is used during pattern matching. If you use pattern matching to identify pages, you must specify a pattern that is unique to each fingerprint.
- [Geometric pattern matching](#)
Geometric pattern matching uses graphical images, like page registration marks or vendor logos. You can use geometric pattern matching to identify pages and correct registration problems.
- [Text-based pattern matching](#)
Text-based pattern matching works much like geometric pattern matching except that the anchor objects are text strings. You set up anchor fields in the document hierarchy and then define the anchor zones on each fingerprint.
- [TravelDocs: Use geometric pattern matching to identify pages](#)
You can update the TravelDocs application to use geometric pattern matching to identify pages and correct registration problems.

Parent topic: [Datacap application development](#)

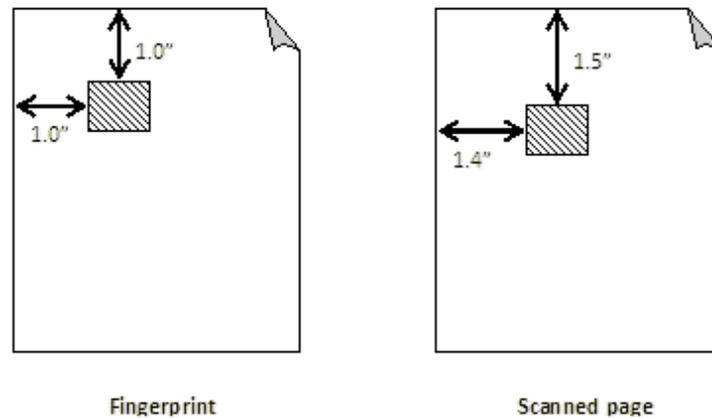
Pattern matching overview

Pattern matching uses anchor objects that you define on the page fingerprints. These anchor objects can be geometric patterns, like a page registration marks or vendor logos, or text-based patterns.

The anchor objects can act both as identification markers used during page identification and as reference points used during registration or realignment of the image.

The Datacap pattern matching actions analyze the runtime pages and look for geometric or text-based patterns that match the patterns in the page fingerprints. It is not unlike standard fingerprint identification, except that it uses only a selected region of the fingerprint image. However, you can use the difference between the location of the pattern on the fingerprint and its location on the runtime page to correct registration problems.

Whether you use geometric pattern matching or text-based pattern matching, the basic concept is the same and is illustrated in the following example. Here, the cross hatched region is the anchor object. In the fingerprint, the anchor is located 1.0 inches from the top and left edge of the page. In the scanned page, the anchor pattern is 1.5 inches from the top and 1.4 inches from the left edge of the page.



A misalignment of this magnitude almost certainly causes a fingerprint match to fail if you are using one of the standard techniques to match fingerprints. Pattern matching attempts to locate the anchor object and, if successful, computes the offsets that are required to bring the page back into alignment. As such, it can handle much larger registration problems. In the previous example, the image must be moved 0.4 inches to the left and 0.5 inches up, so the required image offset values are -80, -100.

Attention: Datacap processes pages at an effective resolution of 200 x 200 pixels per inch, so 0.4 inches is equivalent to 80 pixels and 0.5 inches is equivalent to 100 pixels.

When you use geometric pattern matching with multiple anchor objects, Datacap can do interpolate realignment. The field positions are adjusted based on their proximity to each of the anchor objects.

Although geometric pattern matching and text-based pattern matching are conceptually the same, the implementations are slightly different and use different actions.

- [Considerations for using pattern matching](#)
Some of the things you might consider for pattern matching include when to use for page identification and good anchor patterns.
- [Auto registration with the FindFingerprint action](#)
When FindFingerprint detects a match, it automatically computes the offsets that are required to correct the image registration. It stores these offsets in the *Image_Offset* variable in the page. The ReadZones action uses the offsets when it sets the runtime field positions.

Parent topic: [Pattern Matching](#)

Considerations for using pattern matching

Some of the things you might consider for pattern matching include when to use for page identification and good anchor patterns.

When is pattern matching a good choice for page identification?

If your application must handle various page types that are similar, standard fingerprint identification (FindFingerprint) might generate mismatches. Pattern matching uses a smaller defined region of the page. You can select an area that is unique to each page type and thus avoid mismatches.

What makes a good anchor pattern?

For pattern matching to work, you must have a good anchor pattern. If you are using geometric pattern matching, the image must be composed of simple, solid regions. Images that contain different degrees of shading do not work well for geometric pattern matching.

For text-based pattern matching, the only requirement is that the text pattern is unique to the page type.

What types of pages typically require pattern matching for image registration?

The standard Fingerprint Matching action, FindFingerprint, can tolerate some minor page misalignment relative to the fingerprint image. For more information, see [Auto registration with the FindFingerprint action](#). Pages that have more serious alignment problems or have inconsistently proportioned areas require pattern matching for registration. Faxed images are especially vulnerable because the sending and receiving fax machines might pull the paper through at different speeds, resulting in longer, or shorter images. Additionally, pages with OMR fields require accurate registration, especially if the boxes are closely spaced.

Is there a way to register pages manually?

In certain situations, you might want to do image registration manually. You can register images by using the AIndex web client. For details, see [Manual page identification and registration](#).

Parent topic: [Pattern matching overview](#)

Auto registration with the FindFingerprint action

When FindFingerprint detects a match, it automatically computes the offsets that are required to correct the image registration. It stores these offsets in the *Image_Offset* variable in the page. The ReadZones action uses the offsets when it sets the runtime field positions.

The following example shows the runtime data for a page that Datacap identified correctly and registered automatically by using FindFingerprint.

```
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev027_TM00
0003">
  <V n="TYPE">Air_Ticket</V>
  <V n="STATUS">49</V>
  <V n="IMAGEFILE">tm000003.tif</V>
  <V n="ScanSrcPath">c:\datacap\traveldocs\images\problem_images_page_3.tif</V>
  <V n="RecogStatus">0</V>
  <V n="Confidence">0.8689438</V>          <!-- Confidence level high enough for a
match
  <V n="Image_Offset">-24,-20</V>        <!-- Offsets calculated automatically by
FindFingerprint
  <V n="TemplateID">566</V>              <!-- Matching fingerprint ID
  <V n="Fingerprint Created">No</V>
</P>
```

FindFingerprint calculates the offsets without using anchor objects. However, the offsets it can handle are small. The CalculateOffset action in the Autodoc library can increase the size that the offsets it can handle. Increasing the size that offsets handle can slow down the matching process.

Library	Action	Description
Autodoc	CalculateOffset	Sets the maximum offset supported when you are matching pages to fingerprints.

Anchor objects setup

An anchor object is a region of a fingerprint image that is used during pattern matching. If you use pattern matching to identify pages, you must specify a pattern that is unique to each fingerprint.

The process for setting up anchor objects is the same for a geometric pattern or a text-based pattern.

The coordinates of the anchor object and other information are stored in the document hierarchy. You must create a field-level object to store the details of the anchor object. Aside from the coordinates of the anchor object, the other key element is the `PatternMatch` variable, which identifies the object as a pattern match anchor. Additionally, you typically set the `STATUS` variable of the anchor object to `-1` so the field is not displayed during verification.

The following example shows the XML definition of an anchor field. The anchor is created as follows:

- In Datacap Studio Document Hierarchy pane, you added a field object that is called `Anchor_Object_1` to each page definition.
- The `PatternMatch` of the anchor object variable is set to `1` to identify the object as an anchor object.
- On the Zones tab, a recognition zone is drawn around the anchor on each of the three fingerprints in this example.

```
<F type="Anchor_Object_1">
  <V n="ID">0</V>
  <V n="TYPE">Field</V>
  <V n="STATUS">-1</V>      <!--STATUS = -1 keeps the anchor field hidden-->
  <V n="Position">0,0,0,0</V>
  <V n="MIN_TYPES">0</V>
  <V n="MAX_TYPES">0</V>
  <V n="ReqConf">8</V>
  <V n="rules"></V>
  <V n="PatternMatch">1</V>  <!--PatternMatch = 1 defines the object as an
anchor-->
  <V n="Pos565">183,195,276,282</V>  <!--Anchor object coordinates for fingerprint
565-->
  <V n="Pos566">636,185,730,284</V>  <!--Anchor object coordinates for fingerprint
566-->
  <V n="Pos567">1094,185,1181,279</V> <!--Anchor object coordinates for
fingerprint 567-->
</F>
```

- [Confidence level setup for pattern matching](#)
The default confidence level for pattern matching is `8`. You can set this value for geometric pattern matching by using the `PatternMatch_*` actions. Or you can set the value for text-based pattern matching by using the `pat_RecogMatch_Id` action.

Parent topic: [Pattern Matching](#)

Confidence level setup for pattern matching

The default confidence level for pattern matching is `8`. You can set this value for geometric pattern matching by using the `PatternMatch_*` actions. Or you can set the value for text-based pattern matching by using the `pat_RecogMatch_Id` action.

Geometric pattern matching

You set required confidence level for geometric pattern matching in the ReqConf variable of the anchor object. You can change the default confidence level by right-clicking the anchor object in the Document Hierarchy pane and choosing Manage Variables.

Alternatively, you can change the ReqConf variable through the Properties pane in the Zone tab.

Text-based pattern matching

The pat_RecogMatch_Id action does not use the ReqConf variable. Instead, it uses the confidence level that is established by using the SetMatchConfidence action.

Library	Action	Description
PatternMatch	SetMatchConfidence	Sets the confidence threshold for pattern matching.

Parent topic: [Anchor objects setup](#)

Geometric pattern matching

Geometric pattern matching uses graphical images, like page registration marks or vendor logos. You can use geometric pattern matching to identify pages and correct registration problems.

The following table identifies some of the key actions in the PatternMatch library that are used for geometric pattern matching.

Library	Action	Description
PatternMatch	PatternMatch_Identify	Identifies a page that uses geometric pattern matching, sets the page type and image offsets, and creates the page data file.
PatternMatch	pat_RegisterZones	Use after you run PatternMatch_Identify if you have multiple anchors on the page. This action adjusts the positions of all fields that are based on the positions of multiple anchor fields.

- [How the PatternMatch_Identify action works](#)
When Datacap starts the PatternMatch_Identify action, it gathers all of the anchor objects from all of the fingerprints, then looks for a match on the current page.
- [Multiple anchor objects](#)
To improve pattern matching accuracy, you can specify multiple anchor objects. For example, by defining anchor objects at the upper left and lower right of the page, you can improve the resulting registration.
- [pat_RegisterZones action to adjust the positions of individual fields](#)
If you are using multiple anchors, you can use the pat_RegisterZones action to compute the optimum offsets for individual fields.

Parent topic: [Pattern Matching](#)

How the PatternMatch_Identify action works

When Datacap starts the PatternMatch_Identify action, it gathers all of the anchor objects from all of the fingerprints, then looks for a match on the current page.

The Datacap system does not search the entire page. Instead, it searches a region 200 pixels greater in each direction than the zone defined for the anchor object. If it finds a match that meets the required confidence

level, it sets the page type and computes the offset values.

Tip: You can change the size of the search region by setting the anchor field's METRIC variable. For example, METRIC=200,300 increases the width by 200 pixels in each direction and the height by 300 pixels in each direction.

The following RRS log entries, which are taken from pageid_rrs.log, illustrate how PatternMatch_Identify works.

```
Created PatternMatch Object
Aquired PM lock
Loading Patterns...
  Vendor_Logo : Pattern Found for '565' with zone (171,194,563,302)
  Vendor_Logo : Pattern Found for '566' with zone (678,191,1044,296) 1
  Vendor_Logo : Pattern Found for '567' with zone (1187,206,1524,286)
Opening 'provider=microsoft.jet.oledb.4.0;data
source=C:\Datacap\TravelDocs\TravelDocsFingerprint.mdb;persist security info=false'
Fingerprint/Rules Database connection established.
Search Image: 'c:\datacap\traveldocs\batches\20110018.013\tm000001.tif'
Matching ID# 566 Conf: 10. 2 X: 240 Y: 271. Search Area: 478,0,1244,496 3 Field
ReqConf:8 --> TRUE
  Calculated offset is: (40,80) 4
Releasing PM lock
```

RRS log entry	Description
1	The PatternMatch_Identify action finds a Vendor_Logo pattern that is defined in each of three fingerprints: 565, 566, and 567.
2	It matches the pattern on fingerprint 566 to a region on the current page with a confidence level of 10 (the highest possible).
3	The search area for fingerprint 566 is 200 pixels greater in each direction than the zone defined for the anchor object.
4	It calculates the x and y offsets between the position of the anchor object and the position of the matching pattern on the page.

Parent topic: [Geometric pattern matching](#)

Multiple anchor objects

To improve pattern matching accuracy, you can specify multiple anchor objects. For example, by defining anchor objects at the upper left and lower right of the page, you can improve the resulting registration.

PatternMatch_Identify does only a simple averaging of the offsets. The pat_registerZones action can do interpolate registration, where field positions are adjusted based on their proximity to each of the anchor objects.

In the following example, Datacap located an anchor object at the upper left and the lower right of the page. Then, it calculated an offset value for each anchor. It averaged these values to determine the offset required to bring the page image into best alignment. It wrote these values to the *Image_Offset* variable of the page.

RRS log

```
Matching ID# 570 Conf: 10. X: 280 Y: 274. Search Area: 0,0,564,513 Field
ReqConf:8 --> TRUE
  Calculated offset is: (100,100) <-- Offset for first anchor object (top
left)
Matching ID# 570 Conf: 10. X: 301 Y: 262. Search Area: 1158,1640,1700,2162
```

```
Field ReqConf:8 --> TRUE
  Calculated offset is: (101,62)      <-- Offset for second anchor object
  (bottom right)
```

Runtime page data

```
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev370
_TM000018">
  <V n="TYPE">Test</V>
  <V n="STATUS">49</V>
  <V n="IMAGEFILE">tm000018.tif</V>
  <V n="ScanSrcPath">c:\datacap\traveldocs\images\refstopbottom.tif</V>
  <V n="RecogStatus">0</V>
  <V n="LC_Confidence">8.571231E-02</V>
  <V n="LC_Image_Offset">-16,0</V>
  <V n="LC_TemplateID">562</V>
  <V n="Fingerprint Created">No</V>
  <V n="Confidence">8.571231E-02</V>
  <V n="TemplateID">570</V>
  <V n="PatternConfidence">10</V>
  <V n="Image_Offset">-100,-81</V>      <-- Average required offset
  <V n="DATAFILE">tm000018.xml</V>
</P>
```

In addition to storing the page image offset, PatternMatch_Identify also creates a page data file. It also stores the offset of the zone for each anchor in the *Zone_Offset* variable of the field.

```
<V n="Zone_Offset">100,100</V>      <-- Offset for first anchor
```

Parent topic: [Geometric pattern matching](#)

pat_RegisterZones action to adjust the positions of individual fields

If you are using multiple anchors, you can use the pat_RegisterZones action to compute the optimum offsets for individual fields.

PatternMatch_Identify computes the offset for the entire page and stores the value in the *Image_Offset* variable of the page.

```
<V n="Image_Offset">-100,-81</V>
```

It also creates the page data file and stores the offset for each anchor in the *Zone_Offset* variable of the field.

```
<V n="Zone_Offset">100,10</V>      <-- Offset for first anchor
```

Using the pat_RegisterZones action to compute handles situations where the difference between the fingerprint and the runtime image varies across the page.

Library	Action	Description
PatternMatch	pat_RegisterZones	Adjusts the positions of all fields on the current page that is based on the positions of the anchor fields of the page.

Important: The pat_RegisterZones action is an alternative to the ReadZones action. Do not use both on the same page.

The TravelDocs application runs the pat_RegisterZones action immediately after it runs the PatternMatch_Identify action.

The following RRS log entries, which are taken from `pageid_rrs.log`, illustrate how `pat_RegisterZones` works.

```
Created PatternMatch Object
Acquired PM lock
Anchor Anchor_1 found.          (1) Looking for offset...
Expected 1384,313,1529,392
Image_Offset
Zone_Offset 81,100              (2)
Anchor Anchor_2 found.          (3) Looking for offset...
Expected 697,1988,1006,2074
Image_Offset
Zone_Offset 102,101            (4)
Register using 2 anchors
Set Arrival_Date from Position to 411,405,713,484 to 507,506,812,585
Set Departure_Date from Position to 1154,412,1450,488 to 1246,513,1532,589 (5)
Set Total_Cost from Position to 1150,781,1331,860 to 1242,882,1417,961
```

- (1) The `pat_RegisterZones` action locates an anchor object (`Anchor_1`) on the current page.
- (2) The `pat_RegisterZones` action retrieves the zone offset for this field that was computed earlier by `PatternMatch_Identify`.
- (3) The `pat_RegisterZones` action locates a second anchor object (`Anchor_2`) on the current page.
- (4) The `pat_RegisterZones` action retrieves the zone offset for this field that was computed earlier by `PatternMatch_Identify`.
- (5) The `pat_RegisterZones` action uses the two zone offsets to compute the new positions for each of the three data fields on the current page. It uses an algorithm that takes into account the proximity of each field to each anchor zone.

Parent topic: [Geometric pattern matching](#)

Text-based pattern matching

Text-based pattern matching works much like geometric pattern matching except that the anchor objects are text strings. You set up anchor fields in the document hierarchy and then define the anchor zones on each fingerprint.

You can view the properties for any of the text anchor zones on the Properties tab.

The following action supports page identification and image realignment by using text-based pattern matching:

Library	Action	Description
PatternMatch	<code>pat_RecogMatch_Id</code>	Identifies a page by using text_based pattern matching, and sets the page type and image offsets. This action uses the patterns (anchor objects) from all fingerprints in the fingerprint library.

- [How the `pat_RecogMatch_Id` action works](#)
When Datacap runs the `pat_RecogMatch_Id` action, it gathers all the anchor objects from the fingerprint library and looks for a match on the current page.
- [Determine the runtime field positions by using anchor offsets](#)
The `pat_RecogMatch_Id` uses the anchor offsets to determine the `Image_Offset` value it writes to the runtime page file.

- [Field adjustment that is based on multiple anchors](#)

The pat_RegisterZones action does not work on pages that were identified by using pat_RecogMatch_Id. The pat_RecogMatch_Id action does not save the individual anchor field offsets.

Parent topic: [Pattern Matching](#)

How the pat_RecogMatch_Id action works

When Datacap runs the pat_RecogMatch_Id action, it gathers all the anchor objects from the fingerprint library and looks for a match on the current page.

For each anchor object, Datacap searches the current page in a region 400 pixels greater in each direction than the text zone defined in the fingerprint. If it finds a match that meets the required confidence level, it sets the page type and computes the offset values.

Restriction: The METRIC variable does not change the size of the search region that is used by pat_RecogMatch_Id.

The following RRS log entries illustrate how pat_RecogMatch_Id works.

```
Created PatternMatch Object
Acquired PM lock
Opening 'provider=microsoft.jet.oledb.4.0;
  data source=C:\Datacap\TravelDocs\TravelDocsFingerprint.mdb;persist security
info=false'
Fingerprint/Rules Database connection established.
#572, path:'C:\Datacap\TravelDocs\fingerprint\572.cco'
FPZone:'1384,313,1529,392' TXTZone:'1408,334,1498,359' Value:'Room' 1
FPZone:'697,1988,1006,2074' TXTZone:'758,2018,811,2036' Value:'Hotel #3'
-----
ANCHOR TEXT:'Room'--->'R[oO0][oO0]m' METRIC:'400,400' 2
SEARCH AREA:'Hotel #3
Room
Check out Wed Nov 24 2010
speed internet microwave fridge 3
$109 95
$329 85'
Matched Value >>Room<< 4
Check FingerPrintID# 572 Match Confidence: 9. Search Area: 1008,0,1700,759
Offset(-80,-100) 5
-----
ANCHOR TEXT:'Hotel #3' --->
'[Z2][oO0][\(\)iIt11][oO0][\ ]*H[\(\)iIt11][\(\)iIt11][\(\)iIt11][\(\)iIt11]
[oO0]p[\ ]*H[oO0]
[\(\)iIt11]e[\(\)iIt11]s' METRIC:'400,400'
SEARCH AREA:'Hotel #3' 6
Matched Value >>Hotel[SPACE CHARACTER]#3<<
Check FingerPrintID# 572 Match Confidence: 9. Search Area: 358,1618,1211,2200
Offset(-100,-101)
RecogMatch FingerPrint#:572 PAGETYPE:Room_Receipt 7
```

RRS log entry	Description
1	The action finds two anchor zones that are defined in fingerprint 572: Room and Hotel #3.
2	It computes a bounding region 400 pixels greater in each direction than the region (TXTZone) defined in the fingerprint CCO file for the first anchor value (Room).
3	It identifies all the text within that bounding region on the current page.

RRS log entry	Description
4	It locates the anchor value <code>Room</code> within the search region.
5	It computes the offset by comparing the word's position on the page to the position in the fingerprint.
6	It repeats the process for the second anchor value.
7	It sets the page's template ID and type because at least one of the zones matched.

Parent topic: [Text-based pattern matching](#)

Determine the runtime field positions by using anchor offsets

The `pat_RecogMatch_Id` uses the anchor offsets to determine the `Image_Offset` value it writes to the runtime page file.

The following example shows how the field positions are determined.

```
<V n="Image_Offset">-100,-101</V>
```

The value here (-100, -101) is from the same Hotel #3 page in [How the pat_RecogMatch_Id action works](#). In this example, you can see that `pat_RecogMatch_Id` used the offset value from `Text_Anchor_2`.

Important: The `pat_RecogMatch_Id` action does not create a page data file and does not store the individual offset for each anchor field.

Later on, when Datacap runs the `ReadZones` action, it uses the `Image_Offset` value to compute the position of each runtime field, for example:

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev111_Arri
val_Date">
  <V n="TYPE">Arrival_Date</V>
  <V n="Position">511,506,813,585</V>
  <V n="STATUS">0</V>
  etc.
```

Compare the field positions that are defined in the fingerprint, which you can see in the Properties pane, with the field positions in the runtime page file. You can see how Datacap used the offset value (-100, -101) to compute the position of each data field.

Field	Fingerprint 572	Runtime page
Arrival_Date	411,405,713,484	511,506,813,585
Departure_Date	1154,412,1450,488	1254,513,1550,589
Total_Cost	1150,781,1331,860	1250,882,1431,961

Parent topic: [Text-based pattern matching](#)

Field adjustment that is based on multiple anchors

The `pat_RegisterZones` action does not work on pages that were identified by using `pat_RecogMatch_Id`. The `pat_RecogMatch_Id` action does not save the individual anchor field offsets.

Parent topic: [Text-based pattern matching](#)

TravelDocs: Use geometric pattern matching to identify pages

You can update the TravelDocs application to use geometric pattern matching to identify pages and correct registration problems.

To update the TravelDoc application to use geometric pattern matching, you set up the pattern match and anchor objects and update the PageID rule. Then, you can run a batch through the workflow.

- [Setting up the pattern match anchor objects](#)
You can create pattern matching zones for each of the air ticket pages. Use the vendor logo at the top of each page as the anchor object you are trying to match.
- [Updating the PageID rule to use pattern matching](#)
You must update the PageID rule to identify unrecognized pages by using pattern matching.
- [Running a batch through the workflow](#)
You can test the pattern matching function. Datacap provided an extra page image file for an Airline #2 air ticket where the image is offset enough to cause a standard fingerprint match to fail.
- [Reviewing the runtime batch files](#)
After you run the batch through the workflow, you can review the runtime batch files to ensure that they were processed correctly.

Parent topic: [Pattern Matching](#)

Setting up the pattern match anchor objects

You can create pattern matching zones for each of the air ticket pages. Use the vendor logo at the top of each page as the anchor object you are trying to match.

About this task

Before, you can define the zones, you must add a field to the document hierarchy for the anchor object.

Procedure

Setting up the pattern match and anchor objects:

1. Click the Datacap Studio Zones tab.
2. In the Document hierarchy pane, click Lock DCO for editing and expand the Flight document.
3. Right-click the Air_Ticket page and choose Add > Field.
4. Rename the new field Vendor_Logo and then use Up Arrow to move it to the top of the list.
5. Right-click the Vendor_Logo field and choose Anchor field. The field is identified as an anchor object by setting the *PatternMatch* variable to 1.
6. In the Properties pane, set the *STATUS* variable to -1.
7. In the Document hierarchy pane, click Save. Leave the document hierarchy locked for editing.
8. In the Fingerprints pane, expand the Flight class and select the first air ticket page (Airline #1).
9. In the Document hierarchy pane, select the Vendor_Logo field. Use the mouse to draw a bounding box around the vendor logo on the page image.
10. Repeat for the remaining air ticket pages (Airline #2 and Airline #3).
11. In the Document hierarchy pane, click Save and then click Unlock DCO.

Parent topic: [TravelDocs: Use geometric pattern matching to identify pages](#)

Updating the PageID rule to use pattern matching

You must update the PageID rule to identify unrecognized pages by using pattern matching.

About this task

You put the pattern matching function at the end so it runs only if standard fingerprint matching and text matching both fail. You are going to use a new sample image that has offsets large enough to make standard fingerprint matching fail. Most applications use one page identification method, this exercise is for illustration purposes only.

Procedure

To update the PageID rule to use pattern matching:

1. Click the Datacap Studio Rulemanager tab.
2. In the Rulesets pane, select the PageID ruleset and click Lock/Unlock ruleset to lock the ruleset for editing.
3. Expand the PageID ruleset. Then, right-click the PageID rule and choose Add Function.
4. Rename the new function Identify using Pattern Match.
5. Click the Actions library tab.
6. Select and add each of the actions that are shown in the following table to the Identify using Pattern Match function by clicking Add to function. Then, set the action parameters as shown in the table.

Library	Action	Parameter
PatternMatch	PatternMatch_Identify	
rrrunner	rrSet	varSource = GeometricPattern varTarget = @P.MatchType

Important: rrSet ("GeometricPattern", "@P.MatchType") stores the string GeometricPattern in a page variable that is called MatchType within the runtime hierarchy. You can see which pages were identified by which PageID function.

7. Add the rrSet action to the Identify using Fingerprint function and set the action parameter as shown.

Library	Action	Parameter
rrrunner	rrSet	varSource = Fingerprint varTarget = @P.MatchType

8. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and choose Publish Ruleset.

Parent topic: [TravelDocs: Use geometric pattern matching to identify pages](#)

Running a batch through the workflow

You can test the pattern matching function. Datacap provided an extra page image file for an Airline #2 air ticket where the image is offset enough to cause a standard fingerprint match to fail.

Procedure

To run a batch through the workflow:

1. Copy the file OffsetAirTicket.tif into the TravelDocs application images folder. Open the page and compare it to the original Airline #2 ticket. You see that the image is offset by 0.5 inches in both the x and y directions.
2. Click the Datacap Studio Test tab.

3. In the Workflow pane, select the VScan task profile under Main Job
4. Click New to start a new batch.
5. Click Process rules for target object. Wait for the VScan task profile to run, click Advance when it completes.
6. Click Process rules for target object. Wait for the Page ID task profile to run click Advance when it completes.
7. On Runtime batch hierarchy tab, scroll to the bottom and make sure that the last page is identified with type Air_Ticket. Then, select it and confirm that the new Airline #2 ticket is displayed on the Image tab.
8. Click Process rules for target object and click Advance to move the batch through the remainder of the workflow.

Parent topic: [TravelDocs: Use geometric pattern matching to identify pages](#)

Reviewing the runtime batch files

After you run the batch through the workflow, you can review the runtime batch files to ensure that they were processed correctly.

Procedure

To review the runtime batch files:

1. Open the most recent batches folder of the TravelDocs application.
2. Open the file PageID.xml and scroll till end of the file to view the details for the last page.

```
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev230
_TM000015">
    <V n="IMAGEFILE">tm000015.tif</V>
    <V
n="ScanSrcPath">c:\datacap\traveldocs\images\offsetairticket.tif</V>
    <V n="RecogStatus">0</V>
    <V n="LC_Confidence">0.4774427</V>           <!--Confidence level
from fingerprint matching
    <V n="LC_Image_Offset">-24,0</V>
    <V n="LC_TemplateID">569</V>               <!--Closest matching
fingerprint
    <V n="Fingerprint Created">No</V>
    <V n="Confidence">0.4774427</V>
    <V n="TemplateID">566</V>                 <!--Matching
fingerprint based on pattern matching
    <V n="PatternConfidence">10</V>          <!--Confidence level
from pattern matching
    <V n="Image_Offset">-100,-100</V>       <!--Offset relative to
fingerprint 566
    <V n="MatchType">GeometricPattern</V>   <!--Page identified
using geometric
pattern matching
    <V n="DATAFILE">tm000015.xml</V>
</P>
```

In this example, the closest matching fingerprint is fingerprint 569, one of the hotel pages. But the confidence level is only 0.477, which is not enough for a match. This result caused the FindFingerprint action to fail.

After the FindFingerprint action failed, Datacap ran the Identify using Text Match function. However, this function failed because the Airline #2 ticket does not include the word "Car." Datacap then ran the Identify using Pattern Match function and found an exact match with the fingerprint 566, the original Airline #2 fingerprint. The match is exact because the vendor logo, not the data, is identical on both

pages. Open the export text file (C:\TravelDocs\export\batch_id.txt). You can see that Datacap located all of the data fields by using the offsets that it calculated during pattern matching and recognized the data successfully.

```
,U Airline #2,Newark, NJ (EWR),Dayton, OH (DAY),MON JAN 10, 2011,  
Dayton, OH (DAY),Newark, NJ (EWR),THUR JAN 13, 2011,360.56,33.23,393.79
```

Parent topic: [TravelDocs: Use geometric pattern matching to identify pages](#)

Workflow automation, routing, and automatic fingerprint generation

You can configure Rulerunner to monitor the job queue and run background tasks like PageID, Profiler, and Export automatically whenever batches are pending.

The manual workflow processing was linear so far. You moved each batch from task to task in the same sequence, VScan > PageID > Profiler, Export. The one exception was when a batch was diverted out of the standard workflow and into the FixUp task to address document integrity problems. You can also do conditional routing, which can route a batch to a specific job if it requires special handling. The example that is used is manual page identification, which works if the automated identification methods fail to recognize a page.

The last new task you to cover is automatic fingerprint generation. Automatic fingerprint generation is useful when you want operators to add new page types to the fingerprint library automatically, instead of defining them by using Datacap Studio. This way the fingerprint and recognition zones are saved and can be used next time that you need to process a page of the same type.

Then, you can configure the TravelDocs application to use Rulerunner for background task processing. You can also update the application to handle unidentified pages by routing the batch to a new web client interface called ProtoId for manual page identification. You add a function to automatically generate fingerprints for the pages.

- [Use Rulerunner to automate background tasks](#)
You can use Rulerunner to automate the processing of background tasks for pending batches.
- [Conditional branching and splitting to route documents](#)
You can use conditional branching and splitting to route a batch or part of a batch to a separate job.
- [Automatic fingerprint generation](#)
You can add a function to your application to generate fingerprints automatically from unrecognized pages.
- [TravelDocs: Automated background processing with Rulerunner](#)
You can update the TravelDocs application to run automated background processing by using Rulerunner.
- [TravelDocs: Handle document integrity failures](#)
After you implement recognition and validation, you must run the document creation and integrity checking tasks in their own task profile to handle document integrity failures.
- [TravelDocs: Identify pages manually](#)
You can implement another conditional branch to handle the situation where you must identify pages manually.
- [TravelDocs: Generating fingerprints automatically](#)
You configured the TravelDocs application for manual page identification and drew bounding zones around each field to obtain the recognition data. But you did not create a new fingerprint or save the recognition zones.
- [TravelDocs: Splitting a document from the main batch](#)
You can split manually identified pages from the main batch and send them to a supervisor for

fingerprint creation.

Parent topic: [Datacap application development](#)

Use Rulerunner to automate background tasks

You can use Rulerunner to automate the processing of background tasks for pending batches.

Up to now, you initiated each workflow task manually, typically by starting the task from the Datacap Web Client or from the Datacap Studio Test tab.

Production environments want to automate batch processing to the greatest extent possible. To achieve this goal, means identifying background tasks and running them automatically.

Background tasks are tasks that do not require operator intervention. The TravelDocs application has three background tasks:

- PageID - identifies each page and assigns a page type
- Profiler - combines pages into documents, and does recognition and validation tasks
- Export - writes structured data to a data repository

Each of these tasks runs a set of rules, updates the runtime batch hierarchy, and marks the task as complete. The Datacap queuing engine then readies the batch for the next task in the workflow. If the rules detect any problems, for example, the batch failed document integrity checking, the task can raise an error condition. You can divert the task out of the normal processing workflow for special handling.

Routing and exception handling are addressed later. For now, focus on moving a batch through the standard workflow with as little operator intervention as possible.

- [Rulerunner overview](#)
Rulerunner is the Datacap background processing engine.
- [Rulerunner configuration](#)
You must configure Rulerunner to be able to run background tasks without user intervention.
- [Rulerunner operation](#)
Rulerunner is configured to monitor job queues for jobs to run in the background.
- [Rulerunner logging](#)
Rulerunner provides extensive logging options that you can configure by using the Rulerunner Manager.

Parent topic: [Workflow automation, routing, and automatic fingerprint generation](#)

Rulerunner overview

Rulerunner is the Datacap background processing engine.

You can configure Rulerunner to monitor the job queue and start designated background tasks automatically whenever batches are pending.

For example, in the TravelDocs application, the VScan task and the Verify task are manual tasks that require an operator. The other three tasks do not require any manual input, so you can run them in the background under the control of Rulerunner.

Parent topic: [Use Rulerunner to automate background tasks](#)

Rulerunner configuration

You must configure Rulerunner to be able to run background tasks without user intervention.

Configuring Rulerunner is a two-step process:

- Define the tasks that you want to run as background tasks in the Datacap Application Manager.
- Configure the background tasks in the Rulerunner Manager.

You do these steps for the TravelDocs application in the [Defining background tasks in Datacap Application Manager](#) and [Setting up background tasks in Rulerunner Manager](#) topics.

Parent topic: [Use Rulerunner to automate background tasks](#)

Rulerunner operation

Rulerunner is configured to monitor job queues for jobs to run in the background.

Rulerunner runs in the background as a Windows service. Although you can configure Rulerunner to work with multiple Datacap applications, consider the case where it is configured for one application.

Rulerunner monitors the job queue of the application that is looking for jobs it is configured to run. The Datacap Web Client Monitor tab is used to monitor the queue. But the actual job queue is maintained in the queue table in the Datacap engine database. By default, the engine database is a Microsoft Access database in the root of the application folder. For example, the engine database of the TravelDocs application is C:\Datacap\TravelDocs\TravelDocsEng.mdb.

Attention: Datacap supports other database types, including DB2®, Microsoft SQL Server and Oracle.

Rulerunner polls the job queue every 10 seconds and looks for pending jobs. When it finds a pending job, Rulerunner processes the batch. Upon completion, the batch is ready for the next task in the workflow. In this way, you can move batches through any number of sequential background tasks automatically.

Parent topic: [Use Rulerunner to automate background tasks](#)

Rulerunner logging

Rulerunner provides extensive logging options that you can configure by using the Rulerunner Manager.

The Rulerunner log is needed when you use Rulerunner with the TravelDocs application. For more information, see [Enabling Rulerunner logging](#) and [Analyze the Rulerunner log](#).

Parent topic: [Use Rulerunner to automate background tasks](#)

Conditional branching and splitting to route documents

You can use conditional branching and splitting to route a batch or part of a batch to a separate job.

So far, most of the batch processing was done by going through the workflow linearly.

- VScan > PageID > Profiler > Verify > Export

The one exception was when you used a branch to divert a batch out of the standard workflow, between the Profiler task and the Verify task. Then, into the FixUp task to address document integrity problems.

You can also consider conditional branching and splitting and ways to route a batch or a portion of a batch to a separate job.

Remember: You cannot raise condition flags from tasks that are running under the Datacap Web Client. The tasks exit normally.

- [Branching versus splitting](#)
The two basic methods for workflow routing are branches and splits.
- [Condition flags](#)
Branching and splitting are initiated by using a condition flag that is raised during starting of a rule.
- [Defining a condition and the associated action](#)
A task can have any number of conditions that are associated with it, although splitting uses only the first condition. Branching actions can use any of the conditions.
- [Jobs to handle special conditions](#)
When you used branching, you sent batches with document integrity problems to the FixUp job. The FixUp job is generated automatically by the Datacap Application Wizard, so all you had to do was configure branching to use the existing job.

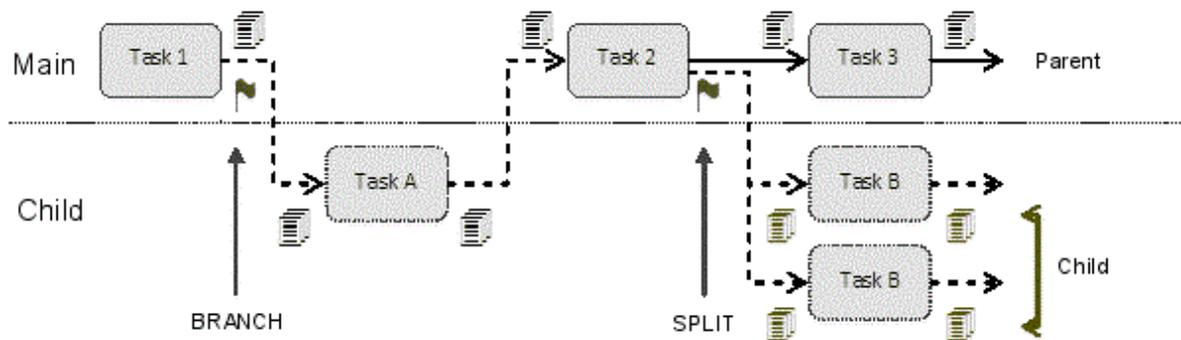
Parent topic: [Workflow automation, routing, and automatic fingerprint generation](#)

Branching versus splitting

The two basic methods for workflow routing are branches and splits.

When you use branching to route workflows, the entire batch is sent from the main job to a child job. When the child job completes, the batch returns to the main job.

When you use splitting to route workflows, documents in the batch are split off from the parent batch and placed into one or more child batches. The child batches are sent to a child job for processing and do not return to the main job.



In this diagram, Task 1 in the main job raises a branch condition and sends the entire batch to Task A in the child job. Task A then returns the batch to Task 2 in the main job. Task 2 in the main job raises a split condition and creates two child batches, which are sent to Task B in the child job. The parent batch continues to Task 3 in the main job.

A batch going through a workflow is tracked by one or more "queue" records, or rows in the Job Monitor. The batch passes sequentially through the tasks in a workflow with a single queue record until a condition is raised. For example, in the TravelDocs application, the Profiler task raises a Document Integrity condition if the sequence of pages and documents does not meet the application requirements. When a task completes successfully after raising a condition, the Datacap Server acts on the condition. The effect of a condition is configured in the workflow as one of the following ("spawn type"):

- Branch

When a task finishes with a Branch condition raised, the Datacap Server creates a new queue record, and routes the batch to the "child job" (Fixup). The original "parent" queue record gets the status as "Waiting". The "Waiting" status means that the processing is suspended on that workflow until the branch workflow finishes successfully. In the TravelDocs Fixup example, after an operator reorganizes or rescans the documents and pages into an acceptable sequence, the Fixup queue status becomes Job Done, and the Main Job picks up processing of the batch where it left off.

- Jump

When a task finishes with a Jump condition, the Datacap Server moves the batch forward or backwards in the existing job workflow by one or more steps, possibly skipping or re-doing some tasks.

- Split

When a task finishes with the Split condition, the Datacap Server creates new "child batches", as many as the application specifies, each with its own BatchID, batch record, and queue record. The original parent batch might continue to the next task in the workflow if any documents remain. The SplitBatch action encapsulates the functionality that is needed to prepare each child batch.

- Stop (or not set)

When a task finishes with the Stop condition, the Datacap Server sets the queue status to Stopped, and no further action is taken.

Parent topic: [Conditional branching and splitting to route documents](#)

Condition flags

Branching and splitting are initiated by using a condition flag that is raised during starting of a rule.

For branching, use the Task_RaiseCondition action to raise the condition flag of the task.

For splitting, use the SplitBatch action to raise the condition flag of the task implicitly.

Branching

In the TravelDocs application, the Batch Document Integrity Check ruleset, under the Document Integrity rule, contains the Batch Route To Fixup function and two related actions, Task_NumberOfSplits and Task_RaiseCondition. In [Document assembly](#), you used the Task_RaiseCondition action to raise a condition flag that determines what happens when the task profile completes. For example, a Batch Route To Fixup function starts only if CheckAllIntegrity returns false.

- The Task_NumberOfSplits action is required and specifies the number of jobs to which the batch is sent before it returns to the main workflow (almost always 1).
- The Task_RaiseCondition action specifies the group index (almost always 0) and condition's index value. For example, the Profiler task can have one condition so the index for this condition is 0.

A batch can be diverted to the Fixup job so that an operator can fix the document integrity problem.

Remember: The branch does not occur until the current task profile is completed.

You update the TravelDocs application to branch when pages require manual identifications.

Splitting

The SplitBatch action implements batch splitting and also raises the condition flag.

Library	Action	Description
Split	SplitBatch	Creates one or more child batches that are based on the value of the specified document-level variable.

Attention: You must run the SplitBatch action at the batch level.

Unlike the branching action, the SplitBatch action does not require a condition index. The implied condition index is always 0 (the first condition). For example, a profile task can have three defined conditions, and the SplitBatch action always raises the first condition, which is Split Condition.

The SplitBatch action uses the document-level variable that is specified in the action parameter. This action determines whether a document is split off to a child batch or remains in the parent batch. In this example, you are using a document variable called *Split*.

```
SplitBatch("@D.Split")
```

This example means any document that has a variable *Split* with any value assigned is split off into a child batch. Furthermore, the value of the *Split* variable determines into which child batch the document goes. So, documents with `<v n="Split">1</v>` go into child batch 1 while documents with `<v n="Split">2</v>` go into child batch 2.

Attention: The values do not need to be numbers. Also, if the value of the variable is the same for all documents, then you get a single child batch.

You implement splitting for the TravelDocs application to split off documents that contain pages that were not recognized during page identification. For more information, see [Updating the Routing ruleset to split the batch](#).

Parent topic: [Conditional branching and splitting to route documents](#)

Defining a condition and the associated action

A task can have any number of conditions that are associated with it, although splitting uses only the first condition. Branching actions can use any of the conditions.

Procedure

To define a condition and the associated action:

1. Start the Datacap Web Client, click the Administrator tab, and click Workflow.
2. Expand the job that contains the task, which includes the condition and associated action, and select the task.
3. In the Selected task details section, select Router from the Mode drop-down menu.
4. In the Parameters section, enter `Return Conditions` under the Key column, and enter the condition name under the Value column.
5. Click Apply.
6. Click the task that you created and in the Selected condition details section, and select or enter values for the corresponding fields, as needed:

Field	Description
-------	-------------

Field	Description
Spawn type:	The following values are available: <ul style="list-style-type: none"> Branch – sends the current batch to the specified job, then returns to the main workflow Jump – skips the next <steps> tasks in the main workflow Split – used with the SplitBatch action to send a child batch to the specified job Stop – stops processing the batch (status = “batch stopped”)
Child Job:	Determines where the batch is sent. Used for Branch or Split only. For example, the batch is sent to Fixup Job.
Parent status:	The batch status when the batch returns. Can be Pending or Hold.
Child status:	Can be Pending or Hold.
Steps:	<ul style="list-style-type: none"> When used with Jump: The number of workflow tasks to jump When used with Branch: The return point after branching is completed (0 = same task, 1 = next task, and so on.)

7. Click Save condition.

For more detailed instructions, see [Adding the conditional branch to the PageID task](#).

Parent topic: [Conditional branching and splitting to route documents](#)

Jobs to handle special conditions

When you used branching, you sent batches with document integrity problems to the FixUp job. The FixUp job is generated automatically by the Datacap Application Wizard, so all you had to do was configure branching to use the existing job.

However, you might need to create a new job to handle a specific condition. For example, if you update the TravelDocs application to identify pages manually. Since there is no existing job to identify pages manually, you must create a job to do it. You can then branch to the new job if the batch contains unidentified pages and then return to the main job when the pages are identified.

A new job requires at least these items:

- A job definition with at least one task
- A program that is associated with each task
- [Creating a job and task](#)
You can create jobs and tasks to run on your workflow.

Parent topic: [Conditional branching and splitting to route documents](#)

Creating a job and task

You can create jobs and tasks to run on your workflow.

Procedure

To define jobs and tasks:

1. On the Datacap Web Client, click the Administrator tab.
2. Open the Workflow page.
3. Create a job.
 - a. Select the workflow for which you want to create a job. Typically, there is only one workflow with the same name as the application.
 - b. Click New and enter the job name.
4. Create a task.
 - a. Click the parent job and click New.
 - b. Define the task by entering values for the following fields:

Field	Description
Name:	The task name
Description:	Description of the task. For example, a Verification task might have a description of <code>Verify with Rule Validation</code> .
Mode	The following modes are available: <ul style="list-style-type: none"> ▪ Batch creation: Typically for Scan or virtual scan (VScan) tasks ▪ Router: Enables job routing (conditional branching) for the corresponding task ▪ Normal: Typically for other tasks where conditions are inapplicable, including such tasks as page identification, verification, and export.
Queue by:	Defines which user and station can process a batch that completes this task. The default is None, which means that there are no restrictions.
Store:	Determines whether to save the user ID or Station ID that completed the task.
Program:	From the drop down menu, select a program, such as Datacap Desktop, Rulerunner, or select a web page (.aspx file) for the task.

- c. Click Setup to specify more options if necessary.
- d. Or click Create Setup to create a default task setup.

Parent topic: [Jobs to handle special conditions](#)

Automatic fingerprint generation

You can add a function to your application to generate fingerprints automatically from unrecognized pages.

You can add automatic fingerprint generation function to the Verify task so that an operator can complete the task. However, you can route a document with unidentified pages to a supervisor for fingerprint generation. By either method, the fingerprint and recognition zones are saved and are used the next time that you encounter a page of the same type.

The following are the basic steps for generating fingerprints automatically:

- Identify the page type, either manually or by using a text-based identification technique.
- Display the page to an operator and have the operator define the recognition zones.
- Use the CreateFingerprint action to create a new fingerprint file from the current page image.
- Use the SetFingerprint action to set the class and type for the new fingerprint.
- Use the iloc_SetZones action to add the recognition zone position information to the document hierarchy.

The following are the new actions:

Library	Action	Description

Library	Action	Description
Auto Doc	CreateFingerprint	Creates a fingerprint from the current page. The fingerprint consists of two files: the image (TIFF) file and the fingerprint processing (CCO) file.
Auto Doc	SetFingerprint	Sets the new fingerprint's class and type.
Intel locate	iloc_Set Zones	Writes the recognition zone coordinates from the current page data file to the Position properties of the corresponding field objects in the document hierarchy XML file.

For details about configuring the TravelDocs application to generate fingerprints automatically, see the topic [Creating the AutoFingerprint ruleset](#).

Parent topic: [Workflow automation, routing, and automatic fingerprint generation](#)

TravelDocs: Automated background processing with Rulerunner

You can update the TravelDocs application to run automated background processing by using Rulerunner.

To update the TravelDoc application to use automated background processing, you must define the background tasks and set up these tasks in Rulerunner Manager. Then, you can enable logging, set up the Job Monitor and run a batch through the workflow.

- [Defining background tasks in Datacap Application Manager](#)
To automate background processing for TravelDocs, you must first define the background processing tasks in Datacap Application Manager.
- [Setting up background tasks in Rulerunner Manager](#)
After you define the background processing tasks, you must first set up these tasks in Rulerunner Manager.
- [Enabling Rulerunner logging](#)
You can enable Rulerunner logging to create log files from which you can monitor processing and troubleshoot problems.
- [Setting up the Job Monitor](#)
You set up the Job Monitor to monitor the status of your batch as the automated background processing tasks run on Rulerunner.
- [Running a batch through the workflow](#)
You can run the batch through the workflow to test the automated background processing results. Use the Job Monitor and the Rulerunner log file to review the results and troubleshoot any problems.
- [Analyze the Rulerunner log](#)
After you run the batch through the workflow, you can analyze the Rulerunner log file to review the results and troubleshoot any problems.
- [Disabling Rulerunner logging](#)
Rulerunner writes to the log file every 10 seconds while it is running. You need to enable logging only at specific times for troubleshooting.

Parent topic: [Workflow automation, routing, and automatic fingerprint generation](#)

Defining background tasks in Datacap Application Manager

To automate background processing for TravelDocs, you must first define the background processing tasks in Datacap Application Manager.

Procedure

To define background tasks in Datacap Application Manager:

1. In the Start menu click IBM Datacap Services> Datacap Application Manager.
2. Select the TravelDocs application and click the Rulerunner tab.
3. Use the Add new task button to create three task profiles - one for each background task:

Task	Task profile
PageID	PageID
Profiler	Profiler
Export	Export

4. Close the Datacap Application Manager.

Parent topic: [TravelDocs: Automated background processing with Rulerunner](#)

Setting up background tasks in Rulerunner Manager

After you define the background processing tasks, you must first set up these tasks in Rulerunner Manager.

Procedure

To set up background tasks in Rulerunner Manager:

1. In the Start menu click IBM Datacap Services> Rulerunner Manager.
2. Click the Rulerunner Login tab.
3. Select Datacap Authentication and enter User ID: admin, Password: admin, and Station ID: 1.
4. Click Connect. If you receive a message that indicates the configuration file does not exist, click Yes to create the configuration file.
5. Click the Workflow: Job: Task tab and select the top-level TravelDocs item. Then, under Main Job, select PageID, Profiler, and Export.
6. Right-click inside the empty right pane and choose Threads > Add Thread.
7. Drag the TravelDocs application Main Job from the left pane onto <thread0> in the right pane.
8. Click Save and then click Yes to create the configuration file.

Parent topic: [TravelDocs: Automated background processing with Rulerunner](#)

Enabling Rulerunner logging

You can enable Rulerunner logging to create log files from which you can monitor processing and troubleshoot problems.

Procedure

To enable Rulerunner logging:

1. In the Rulerunner Manager window, click the Logging tab and then click the Rulerunner Log tab at the bottom of the window.
2. Select the Output to option and leave the target folder set to C:\Datacap.

3. Click Save.
4. Click the Rulerunner Login tab and click Disconnect.
Important: You must be connected to Datacap Server to configure Rulerunner. When you finish configuring Rulerunner, disconnect from the Datacap Server before you start the Rulerunner service.

Parent topic: [TravelDocs: Automated background processing with Rulerunner](#)

Setting up the Job Monitor

You set up the Job Monitor to monitor the status of your batch as the automated background processing tasks run on Rulerunner.

Procedure

To set up the Job Monitor:

1. If the Datacap Web Client is not running, start your web browser and enter the URL
`http://localhost/tmweb.net`.
2. Log in to the Datacap Web Client.
 - a. Select the TravelDocs application.
 - b. In the User ID and Password fields, enter `admin`.
 - c. In the Station field, enter `1`.
3. Click the Monitor tab and select Job Monitor.
4. In the Refresh rate field, right-click and select 10 seconds.

Parent topic: [TravelDocs: Automated background processing with Rulerunner](#)

Running a batch through the workflow

You can run the batch through the workflow to test the automated background processing results. Use the Job Monitor and the Rulerunner log file to review the results and troubleshoot any problems.

Procedure

To run a batch through the workflow:

1. Start Datacap Desktop by clicking IBM Datacap Clients > Datacap Desktop.
2. Log in to Datacap Desktop:
 - a. Enter `TravelDocs` for the application.
 - b. In the User ID and Password fields, enter `admin`.
 - c. In the Station field, enter `1` and click Login.
 - d. In the Shortcut field, select VScan from the menu.
3. In Datacap Desktop, browse to the image files in `../TravelDocs/images`, select a file, and click Scan.
4. When the Datacap Desktop VScan task is complete, click Done.
5. Confirm that the Datacap Web Client Job Monitor and Rulerunner Manager are still running.
6. Arrange the Job Monitor and Rulerunner Manager so that both are visible on your desktop.
7. In Rulerunner Manager, click the Rulerunner tab and click Start.
8. Watch the status of the batch in the Job Monitor. The Task and Status fields go through the following sequence. You might not see each step because of the 10-second refresh interval and the Rulerunner 10-second polling interval.

Sequence	First	Second	Third

Task	PageID (Status pending > running)	Rulerunner (Status pending > running)	Verify (Status pending)
------	-----------------------------------	---------------------------------------	-------------------------

The batch is now ready for verification. Because verification is a manual step, you must complete this task before Rulerunner can run the Export task.

9. Verify the batch as you did before by using Datacap Desktop or the Datacap Web Client.
 - o When you reach the page with the invalid car type, set the type to Other.
 - o If a validation failure message is displayed, click OK to override and continue.
 - o When you reach the end, click OK to finish the batch.
10. Switch to the Job Monitor and watch the status of the batch as Rulerunner runs the Export task.

Task:	Export
Status:	pending > running > Job done

Parent topic: [TravelDocs: Automated background processing with Rulerunner](#)

Analyze the Rulerunner log

After you run the batch through the workflow, you can analyze the Rulerunner log file to review the results and troubleshoot any problems.

Rulerunner generates a separate log file for each active thread. The base product licensing allows single threading only and the log file is C:\Datacap\rulerunner0.log. These excerpts from the Rulerunner log illustrate the sequence of events.

```
ExecuteCode: Grabbed Job[Main Job]:Task[PageID] Queue Id:[42]. <--- Grab batch for
PageID
RunGrabbedQNRelease: BatchID [C:\Datacap\TravelDocs\batches\20110067.005].
RunGrabbedQNRelease: Project Path set
to:[C:\Datacap\TravelDocs\dco_TravelDocs\assemble.set.xml]
RunGrabbedQNRelease: Selected pagefile [rrsvscan.xml]. <--- Read input DCO
file
RunGrabbedQNRelease: Read page file
[C:\Datacap\TravelDocs\batches\20110067.005\PageID.xml]. <--- Create output
DCO file
RunGrabbedQNRelease: RRS run successful.
RunGrabbedQNRelease: ProcessedPages in Batch:[15]
RunGrabbedQNRelease: Job[Main Job]:Task[PageID] queue id: [42] ran
batch[20110067.005]
and the status is [finished]. <--- PageID complete
ReleaseTheQ: Released batch, status is now [pending]. <--- Batch pending
for Rulerunner
```

```
ExecuteCode: Grabbed Job[Main Job]:Task[Profiler] Queue Id:[42]. <--- Grab batch for
Profiler
RunGrabbedQNRelease: BatchID [C:\Datacap\TravelDocs\batches\20110067.005].
RunGrabbedQNRelease: Project Path set
to:[C:\Datacap\TravelDocs\dco_TravelDocs\Profiler.set.xml]
RunGrabbedQNRelease: Selected pagefile [PageID.xml]. <--- Read input DCO
file
RunGrabbedQNRelease: Read page file
[C:\Datacap\TravelDocs\batches\20110067.005\Rulerunner.xml]. <--- Create output
DCO file
RunGrabbedQNRelease: RRS run successful.
RunGrabbedQNRelease: ProcessedPages in Batch:[15]
RunGrabbedQNRelease: Job[Main Job]:Task[Profiler] queue id: [42] ran
batch[20110067.005] and the status is [finished]. <--- Rulerunner
complete
ReleaseTheQ: Released batch, status is now [pending]. <--- Batch pending
for Verify
```

```

ExecuteCode: No batches to process, sleeping for [10] seconds.
ExecuteCode: No batches to process, sleeping for [10] seconds. <--- Monitoring
queue during
ExecuteCode: No batches to process, sleeping for [10] seconds. manual Verify
task
ExecuteCode: No batches to process, sleeping for [10] seconds.

ExecuteCode: Grabbed Job[Main Job]:Task[Export] Queue Id:[42]. <--- Grab batch for
Export
RunGrabbedQNRelease: BatchID [C:\Datacap\TravelDocs\batches\20110067.005].
RunGrabbedQNRelease: Project Path set
to:[C:\Datacap\TravelDocs\dco_TravelDocs\Export.set.xml]
RunGrabbedQNRelease: Selected pagefile [Verify.xml]. <--- Read input DCO
file
RunGrabbedQNRelease: Read page file
[C:\Datacap\TravelDocs\batches\20110067.005\Export.xml]. <--- Create output
DCO file
RunGrabbedQNRelease: RRS run successful.
RunGrabbedQNRelease: ProcessedPages in Batch:[15]
RunGrabbedQNRelease: Job[Main Job]:Task[Export] queue id: [42] ran
batch[20110067.005]
and the status is [finished]. <--- Export complete
ReleaseTheQ: Released batch, status is now [Job done]. <--- Batch at end of
workflow

```

Parent topic: [TravelDocs: Automated background processing with Rulerunner](#)

Disabling Rulerunner logging

Rulerunner writes to the log file every 10 seconds while it is running. You need to enable logging only at specific times for troubleshooting.

Procedure

To disable Rulerunner logging:

1. In the Rulerunner Manager window, click the Rulerunner tab and click Stop. Then, wait for the Rulerunner Service to stop. If you get a timeout message, click OK.
2. Click the Rulerunner Login tab and click Connect.
3. Click the Logging tab and click the Rulerunner Log tab at the bottom of the window.
4. Clear the Output to folder option and click Save.
5. Click the Rulerunner Login tab and click Disconnect.
6. Click the Rulerunner tab and click Start to restart the Rulerunner Service.

Parent topic: [TravelDocs: Automated background processing with Rulerunner](#)

TravelDocs: Handle document integrity failures

After you implement recognition and validation, you must run the document creation and integrity checking tasks in their own task profile to handle document integrity failures.

In the Document Assembly topics, the TravelDocs application completes recognition and validation before it branches to the FixUp job. This routing was not a problem because you did not implement recognition and validation. Now that recognition and validation are implemented, some pages have Status = 1, which indicates low confidence values or validation errors. The Datacap Desktop FixUp task does not finish a job when there are pages with Status = 1.

You must move the document creation task and the integrity checking task out of the Profiler task profile and into their own task profile. Now, you can correct any batch integrity problems before the recognition and validation tasks are run.

- [Moving document creation and integrity checking into the PageID task profile](#)
You can move the document creation and integrity checking tasks into the PageID task profile to handle document integrity failures before recognition and validation are run.
- [Creating the CreateDocs task](#)
The existing Profiler task contains the job routing function that you need for the CreateDocs task, so you must copy and modify the existing task.
- [Configuring Rulerunner to run CreateDocs](#)
CreateDocs is a background task, so you must configure Rulerunner to run it automatically whenever a batch is pending.
- [Running a batch through the workflow](#)
To introduce a document integrity problem, add an optional insurance page to the end of the batch. This orphan page causes the document integrity process to raise an error condition and Datacap routes the batch to the FixUp task.

Parent topic: [Workflow automation, routing, and automatic fingerprint generation](#)

Moving document creation and integrity checking into the PageID task profile

You can move the document creation and integrity checking tasks into the PageID task profile to handle document integrity failures before recognition and validation are run.

Procedure

To move document creation and integrity checking into the PageID task profile:

1. Start Datacap Studio and open the TravelDocs application.
2. On the Rulemanager tab, in the Task Profiles pane, click Unlock task profiles.
3. Expand the Profiler task profile and delete the CreateDocs and Document Integrity rulesets by clicking Remove.
4. Click Add a new task profile, select Custom, enter `CreateDocs`, and click OK.
5. Select the new CreateDocs task profile and add the CreateDocs and Document Integrity rulesets by selecting them in the Rulesets pane and click Add ruleset to profile at the left of the Task Profiles pane.
6. Click Save and then click Lock task profiles.

Parent topic: [TravelDocs: Handle document integrity failures](#)

Creating the CreateDocs task

The existing Profiler task contains the job routing function that you need for the CreateDocs task, so you must copy and modify the existing task.

Procedure

To create the CreateDocs task:

1. In the Datacap Web Client, click the Administrator tab, and click Workflow.
2. Expand Main Job, select Profiler, and click Copy.

3. Change the name of Copy of Profiler to `CreateDocs`.
4. Select the new CreateDocs task and click the up arrow to move the task between the PageID task and the Profiler task.
5. With the CreateDocs task selected:
 - a. Change the Description field to `Create documents`.
 - b. Select Normal as the Mode.
 - c. Leave Queue by and Store set to None.
 - d. Under Parameters, select Rulerunner for the Program and clear any values for any additional parameters if they are present.
 - e. Click Create Setup.
 - f. Click Apply.
6. Click the Shortcuts tab and click New to create a new shortcut.
7. In the Name field, enter `CreateDocs`, and `Create Documents` for the Description.
8. For the Mode, select Manual for Hold.
9. Click Done.
10. Click the Permissions check box. The CreateDocs task is now available on the Datacap Web Client Operations tab.
11. Click Save.

Parent topic: [TravelDocs: Handle document integrity failures](#)

Configuring Rulerunner to run CreateDocs

CreateDocs is a background task, so you must configure Rulerunner to run it automatically whenever a batch is pending.

Procedure

To configure Rulerunner to run CreateDocs:

1. Open the Rulerunner Manager window.
2. If Rulerunner is running, click Stop and wait for the service to stop.
3. Click the Rulerunner Login tab and click Connect.
4. Click the Workflow: Job: Task tab and select the TravelDocs application.
5. Under Main Job, select the CreateDocs task and drag it to <thread0>.
6. Click Save.
7. Click the Rulerunner Login tab and click Disconnect.
8. Click the Rulerunner tab and click Start to restart the service with the new settings.

Parent topic: [TravelDocs: Handle document integrity failures](#)

Running a batch through the workflow

To introduce a document integrity problem, add an optional insurance page to the end of the batch. This orphan page causes the document integrity process to raise an error condition and Datacap routes the batch to the FixUp task.

About this task

After you fix the document integrity problem by deleting the page, processing can continue as normal. Remember: The goal here is to demonstrate the routing capabilities of Datacap. So, deleting the problem page is acceptable. Typically, the corrective action is specified in the business requirements.

Procedure

To run a batch through the workflow:

1. Open C:\Datacap\TravelDocs\images.
2. Delete the files CarRental.tif and OffsetAirTicket.tif (the files that you used for text and pattern matching).
3. Make a copy of Images_Page_02.tif (the first optional insurance page) and name the copy Images_Page_14.tif to create an orphaned insurance page.
4. In the Rulerunner Manager window, confirm that the Rulerunner Service service is running.
5. Start and log in to the Datacap Web Client, and click VScan on the Operations tab.
6. In the VScan window, click Browse, go to C:\Datacap\TravelDocs\images. Then, select a file and click Open, then click Scan.
7. When the VScan task displays a message to indicate that the scanning is complete, click OK, click Done then click OK and Stop.
8. On the Datacap Web Client Operations tab, click Upload.
9. When the Upload task displays a message to indicate that the task is complete, click OK and then click Stop.
10. In the Datacap Web Client, click the Monitor tab to display the Job Monitor page, click Job Monitor to display the job queue.
11. Wait until the Profiler task picks up and finishes the pending batch, and passes the batch to the CreateDocs task.

The CreateDocs task raises a condition flag and routes the batch to the FixUp task (a manual task that Rulerunner cannot process).

12. Click the Operations tab and click Fixup.
13. Select the batch with the FixUp task that is pending, and click Yes in the dialog to confirm that you want to start the selected batch. The batch opens in the Datacap Web Client FixUp window. The last document is selected and the Comments field indicates that the document has an invalid member (the orphaned insurance page).
14. Select the page TM00001 and click Delete. Then, click OK to confirm. Datacap deletes the page and the parent document.
15. Click Finish and then click OK in the Task Finished message box. The FixUp task now has status of Job done and the batch is now pending for the Profiler task.
16. Wait until the Profiler task picks up the pending batch and processes the batch. When Profiler is completed, the batch is pending for the Verify task.
17. Open the pending batch in Datacap Desktop or Datacap Web Client and submit the batch as before. Then, switch to the Job Monitor window and watch the status of the batch as Rulerunner completes the Export task.
18. Delete the file Images_Page_14.tif from the images folder so you do not encounter this problem again.

Parent topic: [TravelDocs: Handle document integrity failures](#)

TravelDocs: Identify pages manually

You can implement another conditional branch to handle the situation where you must identify pages manually.

Currently, the application implements three page identification techniques:

- Fingerprint matching
- Text matching
- Pattern matching

Here, you add another function to the PageID ruleset to manage pages that are still unidentified. If a batch includes unidentified pages, the PageID ruleset raises a condition flag. It sends the batch to the ManualPageID task, where the operator can set the page type manually.

- [Adding a function for manual page identification](#)
You must add the function Identify Manually to configure TravelDocs to do manual page identification.
- [Updating the Recognize Page ruleset](#)
You must update the Recognize ruleset to handle pages that are identified manually.
- [Adding the conditional branch to the PageID task](#)
When the rules to handle manual page identification are complete, you can configure the PageID task. Configure the task to branch to the manual page identification task if there are unidentified pages.
- [Creating the ManualPageID job and task](#)
You must create a job and task to be run on the workflow for manual page identification.
- [Configuring branching and creating a shortcut](#)
You configure branching to route batches to do manual page identification.
- [Configuring the Routing ruleset to handle manually identified pages](#)
You must configure the Routing ruleset to handle manually identified pages.
- [Running a batch through the workflow](#)
Datacap includes an image file for a new air ticket page that you can use to test manual page identification.
- [Recognizing the data on the unidentified page](#)
Airline #4 page is not associated with a valid fingerprint and you did not create any text-based rules for air ticket pages. The new page does not have any recognition data. You recognize the data during verification.

Parent topic: [Workflow automation, routing, and automatic fingerprint generation](#)

Adding a function for manual page identification

You must add the function Identify Manually to configure TravelDocs to do manual page identification.

Procedure

To add a function for manual page identification:

1. In the Datacap Studio Rulesets pane, select the PageID ruleset and click Lock/Unlock ruleset. Then, expand the PageID ruleset to view the two rules.
2. Right-click the PageID rule and choose Add Function. Rename the new function Identify Manually.
3. Add the actions and parameters that are shown in the following table to the PageID > Identify Manually function.

Library	Action	Parameter
rrunner	rrSet	varSource = Manual varTarget = @P.MatchType
DCO	SetPageStatus	1
rrunner	Task_NumberOfSplits	1
rrunner	Task_RaiseCondition	0,0

Attention: Task_RaiseCondition(0,0) references the first condition (index = 0) in the PageID task. You add the condition when you update the Recognize Page ruleset.

4. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and choose Publish Ruleset.

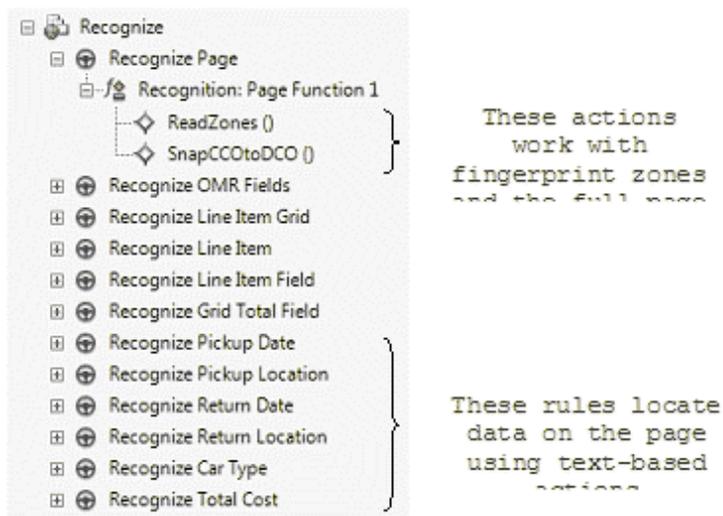
Updating the Recognize Page ruleset

You must update the Recognize ruleset to handle pages that are identified manually.

About this task

Currently, the ruleset works for pages that are identified by using fingerprint matching, text matching, or pattern matching.

- For pages that are identified by using fingerprint matching or pattern matching, you use ReadZone and SnapCCOtoDCO to write the recognition data into the runtime hierarchy.
- For rental agreement pages that are identified by using text matching. You can use various text-based matching actions to locate the recognition data and write it to the runtime hierarchy.



Datacap starts the Recognize Page rule on all page types, even though the rule works only when the current page has a valid matching fingerprint. With manual page identification, there is no matching fingerprint. When fingerprint matching fails, Datacap writes the ID of the closest match into the LC_TemplateID variable of the page. The ReadZones action uses the recognition zones from this low confidence fingerprint even though they are not valid. To work around this problem, you add an action to the Recognize Page rule to enable the rule to exit if the identification method is manual. Later, when you open the page for verification, you define the recognition zones manually.

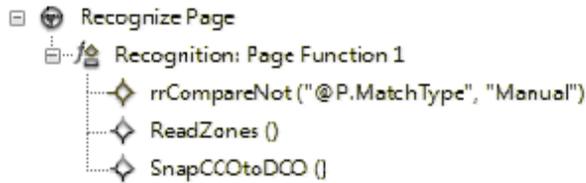
Procedure

To update the Recognize Page ruleset:

1. In the Rulesets pane, select the Recognize ruleset and click Lock/Unlock ruleset. Then, expand the Recognize ruleset and the Recognize Page rule.
2. Add the following action and parameters to beginning of Recognize Page > Recognition: Page Function 1.

Library	Action	Parameter
rrunner	rrCompareNot	object1 = @P.MatchType object2 = Manual

The finished rule is shown in the following example:



If a page has a match type of Manual, the function exits without starting ReadZones.

3. In the Rulesets pane, click Save.
4. Click Lock/Unlock ruleset and choose Publish Ruleset.

Parent topic: [TravelDocs: Identify pages manually](#)

Adding the conditional branch to the PageID task

When the rules to handle manual page identification are complete, you can configure the PageID task. Configure the task to branch to the manual page identification task if there are unidentified pages.

Procedure

To add the conditional branch to the PageID task:

1. Start the Datacap Web Client, select the TravelDocs application, and log in with the Admin account.
2. In the Datacap Web Client, click the Administrator tab and click Workflow.
3. Expand Main job and select the PageID task.
4. In the Selected task details section, select Router from the menu for the Mode.
5. In the Parameters section, enter `Return Conditions` under the Key column, and enter `Page Identification Failed` under the Value column.
6. Click Apply.

Tip: To see the new Page Identification Failed condition, you might need to refresh the page by clicking another tab, and returning to the Workflow page.

You only created the condition node for the PageID task. Before, you can configure the branch, must create the job to use for manual page identification. Complete the steps that are identified in [Jobs to handle special conditions](#).

Parent topic: [TravelDocs: Identify pages manually](#)

Creating the ManualPageID job and task

You must create a job and task to be run on the workflow for manual page identification.

Procedure

To create the ManualPageID job and task:

1. Start the Datacap Web Client, select the TravelDocs application, and log in with the Admin account.
2. In the Datacap Web Client, click the Administrator tab and click Workflow.
3. Select the main TravelDocs node and click New to create a new job node.
4. Enter or select values in the fields for the new job as shown in this table:

Field	Value
Name:	ManualPageID Job

Field	Value
Description:	Manual Page Identification Job
Priority:	1
Parameters:	Leave this field empty.

- Click Apply.
- Select the ManualPageID Job and click New to create a new task.
- Enter or select values in the fields for the new task as shown in this table:

Field	Value
Name:	ManualPageID
Description:	Manual Page Identification task
Mode:	Normal
Queue by:	None
Store:	None
Program:	ProtoID.aspx

- Click Apply.

Parent topic: [TravelDocs: Identify pages manually](#)

Configuring branching and creating a shortcut

You configure branching to route batches to do manual page identification.

Procedure

To configure branching and create a shortcut:

- Start the Datacap Web Client, select the TravelDocs application, and log in with the Admin account.
- In the Datacap Web Client, click the Administrator tab and click Workflow.
- Expand Main job and expand the PageID task.
- Click the Page Identification Failed condition and in the Selected condition details section, enter the following values for the corresponding fields:

Field	Description
Spawn type:	Branch
Child Job:	ManualPageID Job
Parent status:	Pending
Child status:	Pending
Steps:	1

- Click Save condition.
- Datacap Web Client Administrator tab, click Shortcuts and then click New.
- In the Selected shortcut details section:
 - Enter these values for the following fields:

Field	Description
Name:	ManualPageID

Field	Description
Description:	Manual Page Identification
Mode:	Manual for Hold

- b. Under Permissions, click the ManualPageID check box. Confirm that the other check boxes are cleared.
- c. Click Save.

Parent topic: [TravelDocs: Identify pages manually](#)

Configuring the Routing ruleset to handle manually identified pages

You must configure the Routing ruleset to handle manually identified pages.

About this task

You configured the application to display only pages with Status = 1 to the operator. Since manually identified pages have no recognition data, there are no low confidence characters to set the page status to 1. Depending on the way your validation rules are constructed, you might also have no validation errors.

To ensure that manually identified pages are displayed to an operator, you force the page status for the pages to 1. You force the status in the Routing ruleset, which runs immediately after validation.

Procedure

To configure the Routing ruleset to handle manually identified pages:

1. In the Datacap Studio Rulesets pane, select the Routing ruleset and click Lock/Unlock ruleset. Then, expand the ruleset to view the rule.
2. Right-click the Routing Rule 1 rule and choose Add Function. Rename the new function Set Manual Page Status.
3. Use Up Arrow to move the new function to the beginning of the rule.
4. Add the following action and parameters to the Routing Rule 1 > Set Manual Page Status function.

Library	Action	Parameter
rrunner	rrCompare	object1 = @P.MatchType object2 = Manual
DCO	SetPageStatus	1

Important: Make sure that Set Manual Page Status is the first function in the rule.

5. In the Rulesets pane, click Save.
6. Click Lock/Unlock ruleset and select Publish Ruleset.

Parent topic: [TravelDocs: Identify pages manually](#)

Running a batch through the workflow

Datacap includes an image file for a new air ticket page that you can use to test manual page identification.

About this task

Because the image file page fails fingerprint matching, text matching, and pattern matching, Datacap starts the `Identify Manually` function. This function causes the batch to branch to the Manual Page ID task. After you manually identify the page, processing continues as normal.

Procedure

To run a batch through the workflow:

1. Copy the file `NewAirline.tif` from `C:\Datacap\TravelDocs\images\New` to `C:\Datacap\TravelDocs\images\` folder.
2. Confirm that the file `Images_Page_14.tif` is no longer in the images folder. Delete `Images_Page_14.tif`, if necessary.
3. Confirm that Rulerunner Service is running. If necessary, click `Start` on the Rulerunner tab in Rulerunner Manager.
4. In the Datacap Web Client, click `VScan` in the Operations tab. When the task completes, click `OK` and then click `Stop`.
5. On the Datacap Web Client Operations tab, click `Upload`.
6. When the Upload task displays a message to indicate that the task is complete, click `OK` and then click `Stop`.
7. Click the Monitor tab to display the Job Monitor page.
8. Wait until Rulerunner processes the pending batch. The PageID task raises a condition flag and routes the batch to the ManualPageID task.
9. On the Datacap Web Client Operations tab, click the ManualPageID shortcut and wait for the page images to load. Then, scroll till end of the page.

For information on other features in the ProtoId web client, see [Manual page identification and batch restructuring with ProtoId](#).

10. Click the drop-down list beneath the last page and choose `Air_Ticket`.
11. Click `Done` and then click `OK > Stop`.
12. Return to the Job Monitor page. The ManualPageID task now has a status of `Job done` and the batch is now pending for the CreateDocs task.
13. Wait until Rulerunner runs the pending batch through the CreateDocs task and Profiler task. When Rulerunner completes the tasks, the batch is pending for the Verify task.

Parent topic: [TravelDocs: Identify pages manually](#)

Recognizing the data on the unidentified page

Airline #4 page is not associated with a valid fingerprint and you did not create any text-based rules for air ticket pages. The new page does not have any recognition data. You recognize the data during verification.

About this task

You use the Datacap Web Client to recognize the data. But you cannot use Datacap Desktop.

Procedure

To recognize the data on the unidentified page:

1. If necessary, start Datacap Web Client and log in to the TravelDocs application.
2. Click the Verify shortcut to open the pending batch.
3. Go through the batch as you did previously until you reach the Airline #4 page.

4. Click the Outbound_From field, and then, use the mouse to draw a box around the field in the image pane.

When you release the mouse button, the web client inserts the recognition data into the grid.

5. Repeat for the other fields on the page.
6. Click Verify. Datacap displays a message to indicate that the validations passed.
7. Click Submit to submit the page, and then click OK to finish the batch.
8. Click OK and then click Stop.
9. Click the Monitor tab and open the Job Monitor page to watch the status of the batch as Rulerunner runs the Export task.
10. When the export task completes, open the most recent text (.txt) file in the C:\Datacap\TravelDocs\export folder to see the information for the Airline #4 page.

```
,,Okron/Canton, OH (CAK),Washington, DC (DCA),14MAR11,  
Washington, DC (DCA),Okron/Canton, OH (CAK),17MAR11,313.17,64.56,377.73
```

Parent topic: [TravelDocs: Identify pages manually](#)

TravelDocs: Generating fingerprints automatically

You configured the TravelDocs application for manual page identification and drew bounding zones around each field to obtain the recognition data. But you did not create a new fingerprint or save the recognition zones.

Now you can update the application to generate fingerprints for unrecognized pages. When you are finished, run the same batch through the workflow. But this time the application adds a fingerprint to the fingerprint library and the recognition zones to the document hierarchy. The next time that the application encounters an Airline #4 page, the application can use the new fingerprint. Automatic page identification and field recognition is completed by using the new fingerprint.

- [Creating the AutoFingerprint ruleset](#)
You must create the AutoFingerprint ruleset with the actions that enable automatic fingerprint generation.
- [Assigning the rule to each page type](#)
You must assign the Create New Fingerprint rule to each of the page types.
- [Adding the ruleset to the Verify task profile](#)
You must add the AutoFingerprint ruleset to the Verify task profile.
- [Enabling logging for Datacap Web Client](#)
Before you can run a batch through the workflow, you must enable logging for the Verify task.
- [Running a batch through the workflow](#)
You can run a batch through the workflow to test automatic fingerprint generation.
- [Reviewing the RRS log file](#)
If your application did not create the new fingerprint with the zone information, you can check the RRS log file to troubleshoot the problems.

Parent topic: [Workflow automation, routing, and automatic fingerprint generation](#)

Creating the AutoFingerprint ruleset

You must create the AutoFingerprint ruleset with the actions that enable automatic fingerprint generation.

Procedure

To create the AutoFingerprint ruleset:

1. In the Datacap Studio Rulesets pane, right-click the TravelDocs application and choose Add Ruleset.
2. Rename the new ruleset `AutoFingerprint` and rename the default rule from `Rule1` to `Create New Fingerprint`.
3. Select `Function1` and add the actions and parameters in the following table.

Library	Action	Parameter
rrunner	rrCompare	object1 = @P.MatchType object2 = Manual
AutoDoc	SetFingerprintDir	@APPPATH(fingerprint)
AutoDoc	CreateFingerprint	
AutoDoc	SetFingerprint	@D.TYPE,@P.TYPE
Intellocate	iloc_SetZones	

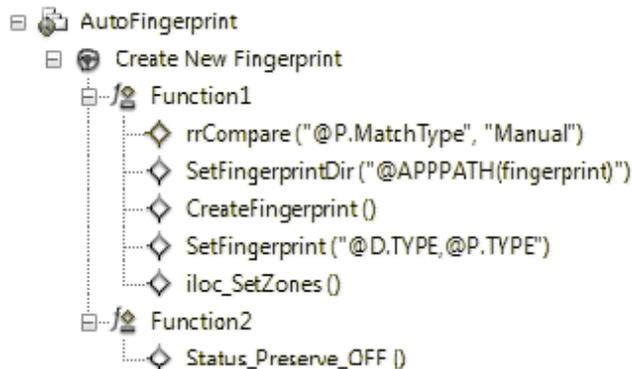
Attention: The parameter on the `SetFingerprint` action sets the fingerprint class to the current document type and the fingerprint type to the current page type. Make sure that `TYPE` is uppercase.

4. Right-click the `Create New Fingerprint` rule and choose Add Function.
5. Select `Function2` and add the following action.

Library	Action	Parameter
rrunner	Status_Preserve_OFF	

The purpose of this function is to ensure that the rule returns True and thus avoid triggering a validation error.

6. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and choose Publish Ruleset.



Parent topic: [TravelDocs: Generating fingerprints automatically](#)

Assigning the rule to each page type

You must assign the `Create New Fingerprint` rule to each of the page types.

Procedure

To assign the rule to each page type:

1. In the Document Hierarchy pane, click Lock DCO for editing.
2. Expand the document hierarchy so you can see all page types.
3. Select the `Rental_Agreement` page type.
4. In the Rulesets pane, select the `Create New Fingerprint` rule and click Add to DCO.
5. In the Document Hierarchy pane, select the `Optional_Insurance` page type and then click Add to DCO.

6. Repeat to add the Create New Fingerprint rule to the Air_Ticket, Room_Receipt, Meals, and Other_Charges pages.
7. In the Document Hierarchy pane, click Save and then click Unlock DCO.

Parent topic: [TravelDocs: Generating fingerprints automatically](#)

Adding the ruleset to the Verify task profile

You must add the AutoFingerprint ruleset to the Verify task profile.

Procedure

To add the ruleset to the Verify task profile:

1. In the Rulesets pane, select the AutoFingerprint ruleset.
2. Click the Task profiles tab and then click Lock/Unlock task profiles.
3. Select the Verify task profile and then click Add ruleset to profile.
4. Expand the Verify task profile and confirm that it contains the Validate ruleset and the AutoFingerprint ruleset.
5. Click Save and then click Lock/Unlock task profiles.

Parent topic: [TravelDocs: Generating fingerprints automatically](#)

Enabling logging for Datacap Web Client

Before you can run a batch through the workflow, you must enable logging for the Verify task.

About this task

You must enable the RRS logging in the Verify task Setup dialog in the Datacap Web Client. You are running the Verify task rules from the Datacap Web Client.

Attention: When you run the Verify task, the log file is saved as verify_rrs.log in the current batch folder.

Procedure

To enable logging for Datacap Web Client:

1. Open the Datacap Web Client, select the TravelDocs application, and login with the Admin account.
2. Click the Administrator tab, and then click Workflow.
3. Expand Main Job, and click the Verify task.
4. In the Selected task details section, click Setup.
5. In the Verify.task.xml - Web Dialog window, scroll down to the Rulerunner Settings section.
6. In the Rulerunner service log: field, enter 3.
7. Scroll down and click Save.

Parent topic: [TravelDocs: Generating fingerprints automatically](#)

Running a batch through the workflow

You can run a batch through the workflow to test automatic fingerprint generation.

Procedure

To run a batch through the workflow:

1. Run a batch through the workflow (see [Running a batch through the workflow](#)), but stop when the batch is pending for verification.
2. Start the Datacap Web Client Datacap client and log in to the TravelDocs application with the Admin account.
3. Click the Verify shortcut in the Operations tab to open the pending batch.
4. Go through the batch as before until you reach the Airline #4 page.
5. Click the Outbound_From field. Then, use the mouse to draw a bounding box around the field in the image pane. When you release the mouse, the web client inserts the recognition data into the grid.
6. Repeat for the other fields on the page.
7. Click Submit. In the background, Datacap runs the AutoFingerprint ruleset to create the new fingerprint file and add the zone information to the document hierarchy. Then, click OK > Stop.
8. In Datacap Studio, click the Zones tab and then click Refresh.
9. Expand the first Flight class (the SetFingerprint action creates a new class even though there is already a class called Flight) and select the new fingerprint.
10. Unlock the document hierarchy and select one of the Air_Ticket fields to activate the zones in the Image View pane.
11. After you review the zones in the Image View pane, lock the document hierarchy.

Parent topic: [TravelDocs: Generating fingerprints automatically](#)

Reviewing the RRS log file

If your application did not create the new fingerprint with the zone information, you can check the RRS log file to troubleshoot the problems.

Procedure

To review the RRS log file:

1. Open the current batch folder.
2. Open the file verify_rrs.log.
3. Scroll to till end of the page of the file to see the log entries for the AutoFingerprint ruleset.

```
ruleset name="AutoFingerprint"
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev229
_14" target object="P" target
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev229
_TM000014"
target type="Air_Ticket"
dco open tag="P"
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev229
_TM000014" type="Air_Ticket"
rule "Create New Fingerprint"
                                func "Function1"
                                action rrCompare
                                ("@P.MatchType", "Manual")
                                action returned true
                                <--Match type is "Manual"
                                /action
                                action SetFingerprintDir
                                (false, false, "@APPPATH(fingerprint)")
                                load rrx code:
"c:\datacap\rrs\autodoc.rrx"
```

```

                                                                    /load
                                                                    action returned true

<--Fingerprint directory established
                                                                    /action
                                                                    action CreateFingerprint
                                                                    action returned true
                                                                    action returned true

                                                                    /action
                                                                    action SetFingerprint
                                                                    action returned true

<--Set the fingerprint class and page type
                                                                    /action
                                                                    action iloc_SetZones
                                                                    load rrx code:
                                                                    /load
                                                                    action returned true

                                                                    /action
                                                                    func result: "true"

                                                                    /func
                                                                    rule result: "true"

/rule
/ruleset
c:\datacap\RRS\Logs\wrrs
end log to batch

```

Parent topic: [TravelDocs: Generating fingerprints automatically](#)

TravelDocs: Splitting a document from the main batch

You can split manually identified pages from the main batch and send them to a supervisor for fingerprint creation.

- [Updating the Routing ruleset to split the batch](#)
You use the existing Routing ruleset to split the batch by using a document-level variable that you created for this purpose.
- [Assigning the Batch Splitting rule to the Close element of the batch](#)
The SplitBatch action must run at the batch level. But it depends on the status of the *Split* variable that in this case is created by a page-level rule in the same ruleset.
- [Routing the split document to a supervisor](#)
Before, you can configure the split condition, you must to create the supervisor job to handle the child batch. Then, you can configure the job router and create the shortcuts for the supervisor job.
- [Running a batch through the workflow](#)
You can run a batch through the workflow to test the document split from the main branch.

Parent topic: [Workflow automation, routing, and automatic fingerprint generation](#)

Updating the Routing ruleset to split the batch

You use the existing Routing ruleset to split the batch by using a document-level variable that you created for this purpose.

About this task

You must set this variable for any document that contains a manually identified page. The Routing ruleset runs at the end of the Profiler task profile after recognition and validation tasks are completed.

Procedure

To update the Routing ruleset to split the batch:

1. In the Datacap Studio Rulesets pane, select the Routing ruleset and click Lock/Unlock ruleset.
2. Expand the Routing ruleset, Routing Rule 1, and the Set Manual Page Status function.
3. Select the Set Manual Page Status function and add the following action and parameters to the end of the function.

Library	Action	Parameter
rrunner	rrSet	varSource = Yes varTarget = @D.Split

This function assigns the value `Yes` to a document-level variable called `Split`, and creates it if necessary.

4. Right-click the Routing ruleset and choose Add Rule. Rename the new rule Batch Splitting.
5. Expand the Batch Splitting rule, select Function1, and add the following action and parameter.

Library	Action	Parameter
Split	SplitBatch	@D.Split

This action raises the condition flag and creates a child batch that contains any documents with the document-level variable `Split`. In this case, the only possible value of the variable is `Yes`. So, all documents are sent to the same child batch.

6. In the Rulesets pane, click Save. Then, click Lock/Unlock ruleset and choose Publish Ruleset.

Parent topic: [TravelDocs: Splitting a document from the main batch](#)

Assigning the Batch Splitting rule to the Close element of the batch

The `SplitBatch` action must run at the batch level. But it depends on the status of the `Split` variable that in this case is created by a page-level rule in the same ruleset.

About this task

In an earlier section that describes the order of rule execution, you learned that batch-level rules typically run before page-level rules. (See [Order of rule execution](#).) However, you also saw how you can attach a rule to the Close element. The rule runs after Datacap finished processing all lower-level objects.

Procedure

To assign the Batch Splitting rule to the Close element of the batch:

1. In the Document Hierarchy pane, click Lock DCO for editing.
2. Expand the document hierarchy so you can see the Close element of the batch.
3. Select the Close element of the batch.

4. In the Rulesets pane, select the Batch Splitting rule and click Add to DCO.
5. In the Document Hierarchy pane, click Save and then click Unlock DCO.

Parent topic: [TravelDocs: Splitting a document from the main batch](#)

Routing the split document to a supervisor

Before, you can configure the split condition, you must to create the supervisor job to handle the child batch. Then, you can configure the job router and create the shortcuts for the supervisor job.

- [Creating the supervisor job](#)
You must create a supervisor job to handle the child batch that you want to split from the main batch.
- [Configuring the job router](#)
After you create the supervisor job, you must configure the job router that is used to send the documents to the supervisor.
- [Configuring the supervisor shortcuts](#)
After you create the supervisor and configure the job router to send the documents to the supervisor, you can configure shortcuts for the supervisor.

Parent topic: [TravelDocs: Splitting a document from the main batch](#)

Creating the supervisor job

You must create a supervisor job to handle the child batch that you want to split from the main batch.

Procedure

Creating the supervisor job:

1. Start the Datacap Web Client, and log in to the TravelDocs application with the `Admin` credentials.
2. Click the Administrator tab, and then click Workflow.
3. Select the TravelDocs workflow and click New to create a new job node.
4. Name the new job node Supervisor Job and enter a description.
5. Select a Priority of 1 and click Apply.
6. Select Supervisor Job and click New.
7. Name the new task node Verify. Datacap populates the other fields in the Selected task details section.
8. Click Apply.
9. Select Supervisor Job again and click New.
10. Name the new task node Export. Datacap populates the other fields in the Selected task details section.
11. Click Apply.

Parent topic: [Routing the split document to a supervisor](#)

Configuring the job router

After you create the supervisor job, you must configure the job router that is used to send the documents to the supervisor.

Procedure

To configure the job router:

1. Open the Datacap Web Client, log in to the TravelDocs application with the `Admin` credentials, and click the Administrator tab.
2. Click Workflow, expand Main Job, and expand the Profiler task.
3. Select the existing Document Integrity Failed condition and click Remove.

This function was moved to the CreateDocs task earlier, so it is no longer needed here.

Tip: If you want to refresh the page, click another tab and return to the Workflow page to confirm that the condition is removed.

4. Select the Profiler task, and under Parameters in the Selected task details section, enter `Split Condition` in the Value column for Return Conditions.
5. Click Apply.
6. Refresh the page, expand the Profiler task, and select Split Condition.
7. In the Selected condition details section, configure the values as follows:

Field	Value
Spawn type	Split
Parent Status	Pending
Steps	1
Child Job	Supervisor Job
Child Status	Pending

8. Click Save condition.

Parent topic: [Routing the split document to a supervisor](#)

Configuring the supervisor shortcuts

After you create the supervisor and configure the job router to send the documents to the supervisor, you can configure shortcuts for the supervisor.

Procedure

To configure the supervisor shortcuts:

1. In the Datacap Web Client, click the Administrator tab, and then click Shortcuts .
2. Click New to create a new shortcut.
3. In the Selected shortcut details section, enter or select these values for the following fields:
 - a. Name: `Supervisor Verify`
 - b. Description: `Supervisor verification`
 - c. Mode: Manual for Hold.
 - d. Under Permissions, clear the check boxes and click the Verify check box under Supervisor Job.
4. Click Save in the Selected shortcut details section.
5. Click New to create a new shortcut.
6. In the Selected shortcut details section, enter or select these values for the following fields:
 - a. Name: `Supervisor Export`
 - b. Description: `Supervisor export data`
 - c. Mode: Manual for Hold .
 - d. Under Permissions, clear the check boxes and click the Export check box under Supervisor Job.
7. Click Save in the Selected shortcut details section.

Parent topic: [Routing the split document to a supervisor](#)

Running a batch through the workflow

You can run a batch through the workflow to test the document split from the main branch.

About this task

Before, you run a batch through the workflow, you need to delete the new Airline #4 fingerprint. So you can process it again as an unrecognized page.

Procedure

To run a batch through the workflow:

1. In Datacap Studio, click the Zones tab.
2. Expand the first Flight class and select the Airline #4 fingerprint.
3. Check the Image View pane to confirm that the Airline #4 fingerprint is selected, and then click Remove selected.
4. Start the Datacap Web Client, and log in to the TravelDocs application with the `Admin` credentials.
5. Click VScan on the Operations tab, browse to the location of the image files, click open and click Scan. When the task completes, click OK and Done. Then, click OK and Stop to return to the Operations tab.
6. Click Upload. When the task completes, click OK and Stop to return to the Operations tab.
7. Start Datacap Desktop, log in to TravelDocs, and select the PageID shortcut. When the task completes, click OK, and exit from Datacap Desktop.
8. Click ManualPageID on the Datacap Web Client Operations tab and wait for the page images to load.
9. Scroll to the bottom and set the page type for the last page to `Air_Ticket`.
10. Click Done, click OK and then click Stop.
11. Click CreateDocs on the Operations tab.
12. When CreateDocs is complete, click Stop.
13. Start Datacap Desktop, log in to TravelDocs, and select the Profiler shortcut. When the task completes, click OK, and exit from Datacap Desktop.
14. Click the Monitor tab and check the Job Monitor to see the result of the split. The child job is pending for the Supervisor Verify task and the main job is pending for the Main Verify task.
15. Click Supervisor Verify on the Operations tab to open the pending batch. The Airline #4 page is displayed.
16. Define the zone for each field. For more information, see [Running a batch through the workflow](#).
17. Click Submit and then click OK. In the background, Datacap runs the AutoFingerprint ruleset to create the new fingerprint file and add the zone information to the document hierarchy.
18. Click OK and Stop.
19. Click the Monitor tab and check the Job Monitor. See the child job that is pending for the Supervisor Export task and the main job still pending for the Main Verify task.
Attention: If you run another batch through the workflow, the batch runs from end to end with no branching or splitting. The Airline #4 page is now in the fingerprint library. If you want to run another batch with branching and splitting, delete the Airline #4 fingerprint from the Datacap Studio Zones tab.

Parent topic: [TravelDocs: Splitting a document from the main batch](#)

Datacap Web Client and remote scanning

You can now update your application by using Datacap Web Client Administrator and run a batch through the entire workflow by using a combination of web components and Rulerunner.

The Datacap Web verification client is discussed in other topics and the manual page identification client topics. These topics describe some of the other Datacap Web Client components, including the remote scanning client, other verification clients, and the Datacap Web Client administration interface.

- [Moving the workflow to Datacap Web Client](#)
The batch processing workflow that you developed so far is a mix of Datacap Desktop tasks, web-based tasks, and Rulerunner background tasks.
- [Scanning images remotely](#)
The Datacap Web Client remote scanning client (scancl.aspx) and the related upload client (uplbfc.aspx) allow operators to scan and upload batches from remote web clients. When the remote batch is queued on the server, Datacap processes the batch in the same manner that it processes any other job.
- [Remote virtual scanning](#)
The Datacap Web Client virtual scanning client (vscancl.aspx) is similar to the remote scanning clients. However, instead of scanning files, the virtual scanning client imports them from a local image folder. You then upload the batches to integrate them into the regular workflow.
- [Verification by using the VeriFine web client](#)
The VeriFine web verification client (VeriFine.aspx) generates verification pages that are automatically based on the document hierarchy.
- [Verification, page identification, and registration by using AIndex](#)
The AIndex web client (aindex.aspx) is similar to VeriFine, but includes full support for multipass verification and manual image registration. As with the other web clients, you enable the client by specifying the name of the .aspx file in the Selected task details, Setup Web Dialog. You access this dialog box through the Datacap Web Client Administrator tab.
- [Verification by using the AVerify web client](#)
The AVerify client is functionally similar to the VeriFine client, but lacks the batch restructuring features of VeriFine. Unlike VeriFine, AVerify completes most of the processing on the client, rather than on the Datacap Web Client server.
- [Verification by using the ImgEnter web client](#)
The ImgEnter (imgenter.aspx) web client is different from the other web verification clients in that you enter data through the page image view.
- [Manual page identification and batch restructuring with ProtoId](#)
You use the ProtoId web client (ProtoId.aspx) to do manual page identification. You can use the list beneath each thumbnail to change the current page type. Additionally, the small toolbar above each thumbnail image provides batch restructuring functions.
- [Administering an application](#)
You use the Administrator tab in the Datacap Web Client for all administration tasks. You use this tab to configure your application from any machine on the network.
- [Job monitoring](#)
You can use Monitor tab in the Datacap Web Client to monitor the status of the job queue.
- [TravelDocs: Scanning from Datacap Web Client](#)
You can create, configure, and run tasks that are related to scanning batches remotely by using Datacap Web Client.
- [TravelDocs: Using AIndex for manual page identification and registration](#)
You must be running Datacap 8.0.1 Fix Pack 1 or later to complete this section.

Parent topic: [Datacap application development](#)

Moving the workflow to Datacap Web Client

The batch processing workflow that you developed so far is a mix of Datacap Desktop tasks, web-based tasks, and Rulerunner background tasks.

Datacap includes web components that you can use to administer most of the workflow from a web browser, including the functions that are listed in the following table.

Function	Web page
Remote scanning and image upload	scancl.aspx and uplbfc.aspx
Virtual scanning and image upload	Vscancl.aspx and uplbfc.aspx
Verification	verfine.aspx averify.aspx imgEnter.aspx
Verification, manual page identification, and manual registration	aindex.aspx
Manual page identification and fixup	ProtoID.aspx
Application administration	Standard tmweb.net interface
Job monitoring	Standard tmweb.net interface

Parent topic: [Datacap Web Client and remote scanning](#)

Scanning images remotely

The Datacap Web Client remote scanning client (scancl.aspx) and the related upload client (uplbfc.aspx) allow operators to scan and upload batches from remote web clients. When the remote batch is queued on the server, Datacap processes the batch in the same manner that it processes any other job.

About this task

Attention: The Datacap Web Client remote scanning clients support TWAIN scanners only. Datacap supports two remote scanning options:

- Scan the images from the web client directly into the application *batch* folder. This option requires that you share the *batch* folder and give write permission to all client machines.
- Scan the images to a local folder on the web client and then upload them to the application *batch* folder. Although Datacap Web Client initially stores the image files locally, it creates the runtime batch file in the server *batch* folder. Datacap Web Client can create the file without requiring you to enable sharing on the *batch* folder.

You specify which option to use in the task setup dialog in the Datacap Web Client, as described in [Configuring the remote scanning client](#). You configure the remote scan task and the upload task later in the [Creating a remote scan task](#) and [Configuring the Upload task](#) topics.

- [Configuring the remote scanning client](#)
To configure the remote scanning client, scancl.aspx, you need to use the Datacap Web Client. By default, the scan task is configured to save the image files locally in C:\Datacap\scan and to use the Upload task. If you want to scan directly to the batches folder of the application, you must share the folder and provide write access to all remote clients.
- [Implementing a start panel](#)
A start panel prompts the operator to enter data before the remote scan page is displayed. You can use a start panel to capture any information specific to the batch that you want to collect (for example, date, and operator name).

Parent topic: [Datacap Web Client and remote scanning](#)

Configuring the remote scanning client

To configure the remote scanning client, `scancl.aspx`, you need to use the Datacap Web Client. By default, the scan task is configured to save the image files locally in `C:\Datacap\scan` and to use the Upload task. If you want to scan directly to the batches folder of the application, you must share the folder and provide write access to all remote clients.

Procedure

To configure `scancl.aspx`:

1. In the Datacap Web Client, click the Administrator tab, and then click Workflow.
2. Expand Scan Job and then select MyISScan.
3. In the Selected task details section, click Setup.
4. In the setup dialog (MyISScan.set.xml - Webpage Dialog):
 - a. Scroll down to the Scan section.
 - b. Enter a value for the Scan into directory field. The default is `c:\datacap\scan`.
 - c. To enable direct scanning into the batches folder of the application without using the Upload task, select the Local processing check box. If you do not select this option, Datacap scans the images to a local folder on the web client and then uses the Upload task to upload the images to the application batches folder.
 - d. Click Save.

Parent topic: [Scanning images remotely](#)

Implementing a start panel

A start panel prompts the operator to enter data before the remote scan page is displayed. You can use a start panel to capture any information specific to the batch that you want to collect (for example, date, and operator name).

Procedure

To enable a start panel:

1. Start Datacap Web Client, click the Administrator tab, click Workflow, and select the task for which you want to enable the start panel.
2. In the Selected task details pane, click Setup.
3. In the `task.set.xml - Webpage Dialog` window, scroll to the Scan or Disk Scan (VScan) section, and select the Show the Start Batch Panel check box, if it is not already selected. (Clearing the check box disables Start Batch Panel.)
4. Scroll down and click Save. The start panel displays a data entry field for each batch level field that is defined in the document hierarchy. A batch level field is generally at the same level of documents that are defined in the batch hierarchy. To capture the name of the person who is completing a scan, you can create a batch level field that is called Name in the document hierarchy. In the TravelDocs application, the Name field is at the same level as the Car_Rental, Flight, and Hotel document types. At the beginning of a scanning task, Datacap displays a dialog that prompts the operator to enter a name. In the following runtime batch hierarchy XML sample, Datacap stored the data in a batch level field (`<Field id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev174_Name">`). The ASCII code values in the character elements spell Henderson, which is the name that the operator entered.

```

<B
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev174_20110154.008">
  <V n="TYPE">TravelDocs</V>
  <V n="STATUS">73</V>
  <V n="ScanOperator">admin</V>
  <V n="ScanStation">1</V>
  <D
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev174_20110154.008.01">
  <V n="TYPE"></V>
  <V n="STATUS">0</V>
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev174_TM000001">
  <V n="imagePath">c:\datacap\scan\20110154.008\tm000001.tif</V>
  <V n="TYPE">Other</V>
  <V n="STATUS">49</V>
  <V n="ScanSrcPath">C:\Datacap\TravelDocs\images\Images_Page_01.tif</V>
  </P>
  etc.
  </D>
  <F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev174_Name">
  <V n="TYPE">Name</V>
  <V n="Position">0,0,0,0</V>
  <V n="STATUS">0</V>
  <C cn="10" cr="0,0,0,0">72</C>
  <C cn="10" cr="0,0,0,0">101</C>
  <C cn="10" cr="0,0,0,0">110</C>
  <C cn="10" cr="0,0,0,0">100</C>
  <C cn="10" cr="0,0,0,0">101</C>
  <C cn="10" cr="0,0,0,0">114</C>
  <C cn="10" cr="0,0,0,0">115</C>
  <C cn="10" cr="0,0,0,0">111</C>
  <C cn="10" cr="0,0,0,0">110</C>
  </F>
</B>

```

- [Populating drop-down lists on a start panel](#)
As with verification panels, you can use dictionaries or `SELECT` variables to populate start panel fields.
- [Running validation rules](#)
You can run validation rules on start panel fields.

Parent topic: [Scanning images remotely](#)

Populating drop-down lists on a start panel

As with verification panels, you can use dictionaries or `SELECT` variables to populate start panel fields.

Procedure

- For example, if you create a dictionary of operator names and associate it with the batch level field, the operator can select his or her name from a drop-down list:
- Alternatively, you can use the field's `SELECT` variable to populate the drop-down list from a database. :
For example, the following `SELECT` value gets a list of operator names from the application's lookup database:

```
<SQL flist='Name' dsn="*/lookupdb:cs">SELECT Operator FROM Operators</SQL>
```

Running validation rules

You can run validation rules on start panel fields.

Procedure

To run validation rules on start panel fields, enter the name of the validation task profile in the set.xml file of the remote scan task client:

1. Start Datacap Web Client, click the Administrator tab, click Workflow, and select the remote scan task for which you want to specify the task profile.
2. In the Selected task details pane, click Setup.
3. In the *task.set.xml* - Webpage Dialog window, scroll to the Rulerunner settings section.
4. In the Main task profile field, enter a name of the validation task profile, such as `ValidateStartPanel`.
Datacap starts the `ValidateStartPanel` task profile when the user clicks Submit in the start panel.
5. Scroll down and click Save.
6. Implement validation rules in Datacap Studio.
 - a. Create a task profile with an associated ruleset.
 - b. Create a validation rule for each start panel field you want to validate.
 - c. Bind each rule to the associated batch level field. For example, the following rule validates the operator name by using a database lookup command.
 - Validate Start Panel ruleset
 - Validate Operator rule
 - Validate Operator function
 - `SetIsOverrideable("False")`
 - `Status_Preserve_OFF()`
 - `OpenConnection("@APPVAR(*)/lookupdb:cs")`
 - `ExecuteSQL("SELECT Operator FROM Operators WHERE Operator='%s';",Name)`
 - `CloseConnection()`

Remote virtual scanning

The Datacap Web Client virtual scanning client (`vscancl.aspx`) is similar to the remote scanning clients. However, instead of scanning files, the virtual scanning client imports them from a local image folder. You then upload the batches to integrate them into the regular workflow.

As with the remote scanning client, the remote virtual scanning client provides two upload options:

- Scan the images from the web client directly into the application's batches folder (Local Processing check box is checked in the task's Setup dialog).
- Scan the images to a local folder on the web client and then upload them to the application's batches folder (Local Processing check box is not checked in the task's Setup dialog).

You configure the Local Processing setting by selecting the task in the Workflow page on the Administrator tab of the Datacap Web Client.

Verification by using the VeriFine web client

The VeriFine web verification client (VeriFine.aspx) generates verification pages that are automatically based on the document hierarchy.

It can also generate custom verification pages by using predefined static layouts (see [Configuring the VeriFine client](#)).

In addition to the image view and the data entry panel, a VeriFine page includes a toolbar and a batch tree view that you can use to split or join documents, reorder pages, and mark documents or pages for deletion.

Note: If the batch being verified has a page that is not part of a document, Verifine.aspx creates a new document and adds that page to the newly created document. This behavior is specific to Verifine.aspx. If a Verify.aspx in tmWeb or DcDesktop is used for verify, no new document is created, and the page continues to remain in the batch.

- [Restructure the batch by using the batch tree view \(VeriFine\)](#)
The batch tree view displays the type and status of each document and page within the batch. You can use controls to restructure the batch.
- [Configuring the VeriFine client](#)
You can configure the VeriFine client in the Workflow page that you access through the Administrator tab in the Datacap Web Client.
- [Configuring additional VeriFine settings](#)
You can configure additional VeriFine settings.
- [Creating custom pages](#)
The VeriFine client generates a default verification panel for each page type automatically. It maps each of the fields in the document hierarchy into a two-column table. You can use VeriFine to replace the standard verification panel with a custom (static) panel.

Parent topic: [Datacap Web Client and remote scanning](#)

Restructure the batch by using the batch tree view (VeriFine)

The batch tree view displays the type and status of each document and page within the batch. You can use controls to restructure the batch.

UI Control	Description
Check mark (✓) on a folder (Split document icon)	Splits the current document so the selected page becomes the first page in a new document.
Two pages on a folder (Join documents icon)	Joins the current document and the previous document.
Mark for deletion icon (X)	Marks the selected document or page for deletion. Documents are assigned the first Ignored Page Statuses (IPS) value; pages are assigned the second IPS value. (See the section <i>Configuring the page and field status settings</i> in Configuring the VeriFine client .)
Up arrow icon (↑)	Moves up the selected page within the current document.

UI Control	Description
Down arrow icon (↓)	Moves the selected page down within the current document.

Parent topic: [Verification by using the VeriFine web client](#)

Configuring the VeriFine client

You can configure the VeriFine client in the Workflow page that you access through the Administrator tab in the Datacap Web Client.

More settings are described in [Configuring additional VeriFine settings](#).

To configure VeriFine settings:

1. Start the Datacap Web Client.
2. Log in to the application that contains the job and task that use the VeriFine client.
3. Click the Administrator tab, and click Workflow.
4. Expand the job that contains the task that uses VeriFine.aspx, and select the task.
5. Click Setup in the Selected task details section.
6. Scroll to the relevant section in the *task.set.xml* Webpage Dialog, and enter or select values for the corresponding parameters.

Configuring the page layouts

By default, VeriFine uses a generic two-column layout that is based on the document hierarchy for the current page and is generated automatically.

You can use the VeriFine section to assign Image View, Data Entry Panel, and the Batch View to a specific location on the page.

For information on creating custom static pages for use with VeriFine.aspx, see [Creating custom pages](#).

Configuring the page and field status settings

The page and field status settings in the Navigation sections (Ignored Page Status(es), Done Page Statuses, and others) control how VeriFine manages pages and fields with the specified status values. See the *STATUS* variable in the [Standard Variable Reference](#) for a list of commonly used status values.

Setting	Description
Ignored Page Status(es)	<p>Determines which pages to ignore. Datacap does not display a page if it has one of the specified status values. Additionally, the first value is assigned to any document you mark for deletion in the VeriFine tree view pane. The second value is assigned to any page you mark for deletion. Do not include any Validation Statuses or Done Page Statuses value in this list.</p> <p>For example, if you specify 72, 74, Datacap does not display pages with STATUS=72 or 74. Additionally, Datacap assigns STATUS=72 to documents marked for deletion and STATUS=74 to pages marked for deletion.</p>

Setting	Description
Done Page Statuses	<p>Determines when a batch is complete. When all pages have one of the specified values, Datacap displays</p> <p>All documents are complete. Click OK to finish the batch.</p> <p>Otherwise, you can put the batch on hold only.</p> <p>For example, if you specify Done Page Statuses=0,2, the batch is complete when all pages have STATUS=0 (OK) or STATUS=2 (validation failure overridden by the operator).</p>
Ignored Field Statuses	<p>Determines which fields to hide. Datacap does not display a field if it has one of the specified status values.</p> <p>For example, if you specify Ignored Field Statuses =-1, Datacap does not display fields with STATUS=-1 (hidden).</p>
Done Field Statuses	<p>Determines which fields to hide when the Problems only check box is selected. Datacap does not display any field with one of the specified status values.</p> <p>For example, if you specify Done Field Status=0 and the Problems only check box is selected, Datacap does not display fields with STATUS=0 (OK).</p>
Validation statuses	<p>Specifies the status value that is assigned to the current page after validation:</p> <ul style="list-style-type: none"> • The first value is assigned when validation passes (Done) • The second value is assigned when the operator overrides a validation error (Override) • The third value is assigned when validation fails and override is not used (Fail) <p>For example, Validation statuses=0,2,1 specifies Done status = 0; Override status = 2; Fail status = 1.</p>

Parent topic: [Verification by using the VeriFine web client](#)

Configuring additional VeriFine settings

You can configure additional VeriFine settings.

Procedure

1. Start the Datacap Web Client.
2. Log in to the application that contains the job and task that use the VeriFine client.
3. Click the Administrator tab, and click Workflow.
4. Expand the job that contains the task that uses VeriFine.aspx, and select the task.
5. Click Setup in the Selected task details section.
6. Scroll to the relevant section in the *task.set.xml* Webpage Dialog, and enter or select values for the corresponding parameters.
 - a. To load pages, click the Load all documents check box in the Document startup section.
Tip: This option loads all page XML files for the current document into a temporary folder. Use this option for cross-page validations that must access multiple pages within a document. However, this option adversely affects system performance.
 - b. To load images, click the Load all images check box in the Document startup section.
Tip: This option is enabled by default, and loads all images in a document when you open that document. Disable this option when you have large documents that require much time to load all

of the images before the first page is displayed. Enabling this option delays the display of subsequent pages.

- c. To configure the batch tree view, enter values in the DCO Tree View section for Display variables.
Tip: By default Datacap displays the type and status variables for each document and page. To add a variable, click the plus sign (+), or to remove a variable, click the minus sign (-).
- d. To customize background field colors, scroll to the Page Processing section, and in the Background field colors subsection, enter color values for the following field types:

Field	Default color
Low Confidence fields	yellow
Invalid fields	lightpink
Normal fields	white

Tip: X11 color names are supported.

- e. To enable multi-pass verification, enter values in the Alternative or Blind texts section for DCO field alt text index and Blind confirm index. Datacap applications can display the same page to multiple operators to ensure that accurate data entry and verification. The VeriFine client provides limited support for multi-pass verification that includes two-pass verification, whereas the AIndex client provides full support for multi-pass verification, including double-blind. For details, see [Verifying in multiple passes](#).

Parent topic: [Verification by using the VeriFine web client](#)

Creating custom pages

The VeriFine client generates a default verification panel for each page type automatically. It maps each of the fields in the document hierarchy into a two-column table. You can use VeriFine to replace the standard verification panel with a custom (static) panel.

Procedure

To create and use custom pages:

1. Create a custom panel layout with the Custom Layout Generator.
 - a. Open the Datacap Web Client Custom Layout Generator (<http://<server>/tmweb.net/task/support/dragresize.htm>).
 - b. Create a layout by browsing to and selecting an .xml file, or edit an existing layout by browsing to and selecting an .ascx file.
 - c. If you are creating a new layout, select a page in the Page menu, and click Generate.
 - d. Move or resize any labels or fields, as wanted.
 - e. Click Save Panel.

2. Locate the panel layout file (.ascx) that you saved in Step 1.

The file is in one of the following folders, depending on the version of Windows that you are using.

- o C:\ (the root of the C: drive)
 - o C:\Users\<username>\AppData\Local\VirtualStore
3. Rename the file to match the page type. For example, you might name the file Rental_Agreement.aspx.
 4. Move the .ascx file into the C:\Datacap\tmweb.net\Task folder.
Important: You must repeat these steps for each custom page layout.
 5. Specify the custom panel in the task's setup.xml file.
 - a. Start the Datacap Web Client and log in the TravelDocs application with the admin account.
 - b. Click the Administrator tab, and then click Workflow.

- c. Select the Verify task, and click Setup.
 - d. In the Verify.set.xml - Webpage dialog window, scroll to the Custom web panels section and click the Use custom web panels check box.
 - e. Enter a page type and the corresponding .aspx file name, for each of the custom pages that you created.
 Tip: You can mix custom page layouts with default page layouts. For example, if your application has 10 page types but you create only two custom layouts, specify just the two custom layouts. VeriFine uses the default layout for the remaining page types.
 - f. Click Save and close the window.
 Tip: To revert to the default panels at any time, clear the Use custom web panels check box and click Save.
6. Open a batch in the Verify task and confirm that the task is using the new custom panels.

Parent topic: [Verification by using the VeriFine web client](#)

Verification, page identification, and registration by using AIndex

The AIndex web client (aindex.aspx) is similar to VeriFine, but includes full support for multipass verification and manual image registration. As with the other web clients, you enable the client by specifying the name of the .aspx file in the Selected task details, Setup Web Dialog. You access this dialog box through the Datacap Web Client Administrator tab.

For more information, see [Verifying in multiple passes](#) and [Manual page identification and registration](#).

AIndex generates verification pages automatically that are based on the document hierarchy, and includes a toolbar, an image view, a data entry panel, and a batch tree view. The web page is similar to VeriFine, but AIndex does not display image snippets in the data entry panel. If you do not require multipass verification, VeriFine is a better choice for web verification.

- [Restructure the batch by using the batch tree view \(AIndex\)](#)
 The batch tree view displays the type of each page within the batch. You can use this view to change the document or page type and restructure the batch.
- [AIndex client configuration](#)
 You configure the AIndex client in the Selected task details, Setup Web Dialog that you access through Datacap Web Client Administrator tab.
- [Verifying in multiple passes](#)
 Datacap applications can display the same page to multiple operators to ensure that accurate data entry and verification. Datacap supports two main implementations of multi-pass verification: *two pass* and *double blind*. Other implementations are possible, but you focus on these two implementations.
- [Manual page identification and registration](#)
 You can use the AIndex web client to complete manual page identification and manual registration.

Parent topic: [Datacap Web Client and remote scanning](#)

Restructure the batch by using the batch tree view (AIndex)

The batch tree view displays the type of each page within the batch. You can use this view to change the document or page type and restructure the batch.

Table 1. Batch tree view controls

Control	Description
---------	-------------

Control	Description
Start Doc check box	A check mark indicates that the current page is the first page in a document: <ul style="list-style-type: none"> • If the check box is cleared, selecting it splits the current document so the page becomes the first page in a new document. • If the check box is selected, clearing it joins the current document and the previous document.
<ul style="list-style-type: none"> • Hotel <ul style="list-style-type: none"> ◦ R <ul style="list-style-type: none"> o o m – R e c e i p t 	Displays the document type and page type for the current page. You can change the page type. If the page is the first page in a document, you can also change the document type.
Move up button	Moves the selected page up.
Move down button	Moves the selected page down.
Problem fields only Check box	Select to display only problem fields in the data entry panel.
Anchors button	See Manual page identification and registration .

Parent topic: [Verification, page identification, and registration by using AIndex](#)

AIndex client configuration

You configure the AIndex client in the Selected task details, Setup Web Dialog that you access through Datacap Web Client Administrator tab.

The page and field status settings are the same as for the VeriFine client. (See the "Configuring the page and field status settings" section in the [Configuring the VeriFine client](#) topic.)

In addition, AIndex supports the following settings that you can use to implement [Verifying in multiple passes](#):

- DCO field alt text index: Specifies the DCO field alternate text index.
- Blind confirm index: Enables automatic double-blind checking that requires matching another alternate value (specified index) or retyping same value twice.
- Show other alt texts: Specifies whether to display other alternatives as hyperlinks.

Parent topic: [Verification, page identification, and registration by using AIndex](#)

Verifying in multiple passes

Datacap applications can display the same page to multiple operators to ensure that accurate data entry and verification. Datacap supports two main implementations of multi-pass verification: *two pass* and *double blind*. Other implementations are possible, but you focus on these two implementations.

Procedure

- In two-pass verification:
 1. An operator (or a recognition engine) enters the initial value for each field.
 2. Datacap displays the page to a second operator but hides the initial values. The operator enters a new value for each field.

Tip: If you are using a recognition engine to implement the first pass, you might choose to display only low confidence fields to the operator.
 3. For each field, Datacap compares the new value to the initial value. If they match, Datacap accepts the value; otherwise, the operator must reenter the value. Only when the operator enters the same value twice in a row does Datacap accept the value.
- In double-blind verification:
 1. An operator (or a recognition engine) enters the initial data values.
 2. Datacap displays the page to a second operator but hides the initial values. The operator enters a new value for each field and Datacap saves all the values (no comparing).
 3. Datacap displays the page to a third operator. Using a feature of AIndex that displays multiple values, the operator can see both the initial value and the second value.
 4. For fields where the initial value and the second value are different, the operator must determine which value is correct, or enter a new value:
 - Clicking the initial value (or pressing Alt+Shift+A when the field has the focus) moves the initial value into the data entry field.
 - To enter a new value, the operator must enter the same value twice in a row.
- [Storing multiple values in the runtime page data file](#)

Your application can store values from multiple data entry passes in the runtime page data file.
- [Actions that support multi-pass verification](#)

The actions that are required to move and copy data are shown in this table.
- [Settings that support multi-pass verification](#)

You can specify these settings in the Datacap Web Client Administrator tab, clicking Workflow, and selecting the relevant task.
- [Example of two-pass data entry](#)

In this example, *Operator 1* can be a person or a recognition engine. If you use a recognition engine for the initial data entry, it runs as a background process so there is no user interface shown.
- [Example of double-blind data entry](#)

In this double blind example, *Operator 1* can be a person or a recognition engine. If you use a recognition engine for the initial data entry, the recognition engine runs as a background process, in which case there is no user interface.

Parent topic: [Verification, page identification, and registration by using AIndex](#)

Storing multiple values in the runtime page data file

Your application can store values from multiple data entry passes in the runtime page data file.

About this task

Datacap can store multiple values in the runtime page data file for any specified field:

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev197_Vendor">
    <V n="TYPE">Vendor</V>
    <V n="Position">0,0,0,0</V>
    <V n="STATUS">1</V>
    <C n="6,8,10" cr="0,0,0,0">68,49,83</C>
    <C n="6,8,10" cr="0,0,0,0">97,50,112</C>
    <C n="6,8,10" cr="0,0,0,0">116,51,105</C>
</F>
```

In the task example, the Vendor field has three 3-character values that are represented by the ASCII characters shown. The first value (AltText[0] = ASCII 68,97,116 = "Data") is the primary data value.

Your application can use this structure to store values from multiple data entry passes. However, because data is always captured in AltText[0], you need to move the data.

Procedure

For example, to implement double blind:

1. Capture the initial data in AltText[0].
2. Move the data into AltText[1].
3. Display the page to Operator 2 and save the new data in AltText[0].
4. Copy the AltText[0] data into AltText[2].
5. Display the page so Operator 3 can review AltText[1] and AltText[2] and accept either version or enter new data in AltText[0].

Parent topic: [Verifying in multiple passes](#)

Actions that support multi-pass verification

The actions that are required to move and copy data are shown in this table.

Library	Action	Description
DCO	PropagateToAltText	Copies the character values from AltText[0] to the specified position.
DCO	ClearAltText	Clears the character values from the specified position in the field's array.

To implement a move, do a copy (PropagateToAltText) and then delete the original (ClearAltText).

The action that is used to compare AltText[0] and AltText[1] values is VoteFld.

Library	Action	Description
VoteFld	VoteFld	Sets the confidence level on each character to high and the field status to 0 (OK) if the AltText[0] and AltText[1] values match. Sets the confidence levels to low and the field status to 1 (problem) if the values do not match.

To implement double blind verification, use the VoteFld action before the page is displayed to the last operator. If the initial and second values do not match, the action sets the field status to 1 and the field is displayed in

red.



Parent topic: [Verifying in multiple passes](#)

Settings that support multi-pass verification

You can specify these settings in the Datacap Web Client Administrator tab, clicking Workflow, and selecting the relevant task.

In the *task.set.xml* Webpage Dialog, scroll to the Alternative or blind text section, and enter the values for the corresponding fields, as needed.

Field	Value	Description
DCO field alt text index	0, 1 or 2:	Determines which "alt text" is the primary text (defaults to AltText[0]). AIndex displays the primary text in the data entry field.
Show other alt texts	3:	AIndex displays AltText[0] in the data entry field and AltText[1] and AltText[2] beside it.
	2:	AIndex displays AltText[0] in the data entry field and AltText[1] beside it.
	1:	AIndex displays the AltText[0] value initially, but you can toggle back and forth between AltText[1] and AltText[0] using the Alt+Shift+A hot key.
	0 or -1:	AIndex displays the AltText[0] value in the data entry field. AltText[1] is hidden but is used for comparisons.
Blind confirm index	1:	AIndex compares the new value to the initial value and forces the operator to enter a new value twice.
	-1:	The operator can enter any new value. There is no comparison and no need to enter the value twice.

Parent topic: [Verifying in multiple passes](#)

Example of two-pass data entry

In this example, *Operator 1* can be a person or a recognition engine. If you use a recognition engine for the initial data entry, it runs as a background process so there is no user interface shown.

Table 1. Example of two-pass data entry

Person or recognition engine	User Interface	AltText [0]	AltText [1]	AltText [2]	Task settings in set.xml file
Operator 1 (initial data entry)	1	1			Show other alt texts is set to -1. (hide alt text) Blind confirm index is set to -1. (do not compare)
Propagate ToAltText ("1") ClearAlt Text("0")			1 (Moved from AltText [0])		

Person or recognition engine	User Interface	AltText [0]	AltText [1]	AltText [2]	Task settings in set.xml file
Operator 2 (initial state)			1		Show other alt texts is set to -1. (hide alt text) Blind confirm index is set to 1. (compare)
Operator 2 (after data entry)	2	2	1		Show other alt texts is set to -1. (hide alt text) Blind confirm index is set to 1. (compare)

The data for Operator 2 data ('2') is now stored in AltText[0] and the data for Operator 1 ('1') is in AltText[1]. In this case, the compare fails, and the operator must enter the same value twice to override the initial value. Important: You specify the Show other alt texts value and the Blind confirm index value in the set.xml Webpage dialog for the task in Datacap Web Client.

Parent topic: [Verifying in multiple passes](#)

Example of double-blind data entry

In this double blind example, *Operator 1* can be a person or a recognition engine. If you use a recognition engine for the initial data entry, the recognition engine runs as a background process, in which case there is no user interface.

Table 1. Double blind example of data verification

	User Interface	AltText [0]	AltText [1]	AltText [2]	Alternative or blind text settings
Operator 1 (initial data entry)	1	1			Show other alt texts = -1 (hide alt text) Blind confirm index = -1 (do not compare)
PropagateToAltText("1") ClearAltText("0")		1			
Operator 2 (initial state)			1		Show other alt texts = -1 (hide alt text) Blind confirm index = -1 (do not compare)
Operator 2 (after data entry)	2	2	1		Show other alt texts = -1 (hide alt text) Blind confirm index = -1 (do not compare)
PropagateToAltText("2")		2	1		
VoteFld()		2	1	2	

	User Interface	AltText [0]	AltText [1]	AltText [2]	Alternative or blind text settings
Operator 3 (initial state)	2 (1)	2	1	2	Show other alt texts = 2 (show alt text) Blind confirm index = 1 (compare)

Important: You specify the Show other alt texts value and the Blind confirm index value in the set.xml Webpage dialog for the task in Datacap Web Client.

In the preceding example, PropagateToAltText moves the initial data entry value (1) from AltText[0] to AltText[1], and ClearAltText removes the value from AltText[0]. Operator 2 enters a value of 2 and then PropagateToAltText copies the value of 2 from AltText[0] to AltText[3]. The VoteFld action sets the field status to '1' (problem) because AltText[0] and AltText[1] do not match, and AIndex highlights the field. Operator 3 can now do one of three tasks:

- Accept the current AltText[0] value ('2').
- Swap in and accept the AltText[1] value ('1').
- Enter a new value twice. The new value becomes the new AltText[0] value.

The following table shows the result for each of these three actions.

Table 2. Results of the blind example actions

		AltText [0]	AltText [1]	AltText [2]	Alternative or blind text settings
Operator 3 (accept AltText[0])	2 (1)	2	1	2	Show other alt texts = 2 (show alt text) Blind confirm index = 1 (compare)
Operator 3 (use AltText[1])	1 (2)	1	1	2	Show other alt texts = 2 (show alt text) Blind confirm index = 1 (compare)
Operator 3 (enter new data)	3 (1)	3	1	2	Show other alt texts = 2 (show alt text) Blind confirm index = 1 (compare)

Parent topic: [Verifying in multiple passes](#)

Manual page identification and registration

You can use the AIndex web client to complete manual page identification and manual registration.

- You first set the document and page type by using the drop-down menu in the batch tree view pane.
- You then click Anchors at the end of the batch tree view to select the matching fingerprint and complete manual page registration.

When you register the page, Datacap displays a floating image of the anchor object of the page that you align with the actual anchor image on the current page. When you place the anchor object, Datacap computes the required page offsets that are used to calculate the positions of the data fields on the page.

Datacap can usually handle page identification and registration for you automatically, even when the offsets relative to the fingerprint are large (see [Pattern Matching](#)). Using manual page identification and registration might be useful to you in special situations. However, if you are using AIndex for manual page identification, you must run the manual page identification task after the CreateDocs task because AIndex requires a structured batch. You also need to confirm that each unidentified page is a separate document. These latter steps are discussed when you update the TravelDocs application to use AIndex.

- [Enabling manual page registration \(manual anchoring\)](#)
You can enable manual anchoring for a specific page type.
- [Registering a page by using manual anchoring](#)
Manual anchoring is only enabled for pages that include an anchor field with the variable `Required=1` and where the anchor field's position in the runtime page data file is undefined or set to 0,0,0,0.

Parent topic: [Verification, page identification, and registration by using AIndex](#)

Enabling manual page registration (manual anchoring)

You can enable manual anchoring for a specific page type.

Procedure

1. Define an anchor field for the page type and specify the zone position on each of the corresponding page fingerprints. See [Setting up the pattern match anchor objects](#). Confirm that the field's *PatternMatch* variable is set to 1.
2. Add a variable that is called *Required* to the anchor field and set its value to 1.
3. For the task in which you want to perform manual anchoring, specify the location of the application's fingerprint folder. For example, you might enter C:\Datacap\TravelDocs\fingerprint) in the task's Setup dialog in the Datacap Web Client

Parent topic: [Manual page identification and registration](#)

Registering a page by using manual anchoring

Manual anchoring is only enabled for pages that include an anchor field with the variable `Required=1` and where the anchor field's position in the runtime page data file is undefined or set to 0,0,0,0.

About this task

Unidentified pages (pages of type `Other`) typically do not have a runtime page data file. So you must first assign a page type. When you assign a page type, AIndex creates a page data file and sets all of the fields to 0, 0, 0, 0. If the page type you assign has an anchor field with `Required=1`, AIndex enables manual anchoring.

AIndex checks document integrity when you submit the batch. So your documents and pages must be structured correctly. Additionally, you need to have appropriate Done Page Statuses, Validation Statuses, Done Field Statuses, and Ignored Field Statuses values in the task's .set.xml file (see [Configuring the VeriFine client](#)). You do this task later when you update the TravelDocs application to use AIndex (see [Updating ManualPageID](#)).

Procedure

To identify or register a page by using manual anchoring:

1. In the AIndex batch tree view pane, select the page that requires identification or manual registration.

2. If the page is unidentified, use the drop-down lists to set the document type and page type. Depending on the position of the page within the batch, you might need to select the Start Doc option before you can set the document type.
3. If manual anchoring is enabled for the page type that you selected, Datacap displays a message that you must set the anchor position. Click OK to close the message box.
4. Click Anchors in the batch tree view pane.
5. If the page is unidentified, Datacap displays thumbnails of all fingerprints with a required anchor (Required=1). You are asked to double-click the matching fingerprint. Click OK to close the message box and then double-click the matching fingerprint. Datacap reads the position of the anchor object for the fingerprint that you selected from the document hierarchy, and floats a red version of the anchor image over the current page image.
6. Use the mouse to align the anchor object with the page image.
7. When you complete the batch, Datacap writes the anchor position to the runtime page data file and writes the offset values to the runtime batch hierarchy. The following table provides an example.

Runtime page data file	Runtime batch hierarchy file
<pre><P id="_dcs_markdown_workspace_Transform_h tmlout_0_com.ibm.dc.develop.doc_dcdev19 0_appdevguide_TM000006"> <F id="_dcs_markdown_workspace_Transform_h tmlout_0_com.ibm.dc.develop.doc_dcdev19 0_appdevguide_Vendor_Logo"> <V n="TYPE">Vendor_Logo</V> <V n="Position">221,229,608,332</V> <V n="STATUS">-1</V> </F> etc.</pre>	<pre><P id="_dcs_markdown_workspace_Transform_ htmlout_0_com.ibm.dc.develop.doc_dcdev 190_appdevguide_TM000006"> <V n="IMAGEFILE">tm000006.tif</V> <V n="TYPE">Air_Ticket</V> <V n="STATUS">0</V> <V n="Image_Offset">-45,-29</V> <V n="TemplateID">562</V> etc.</pre>

Parent topic: [Manual page identification and registration](#)

Verification by using the AVerify web client

The AVerify client is functionally similar to the VeriFine client, but lacks the batch restructuring features of VeriFine. Unlike VeriFine, AVerify completes most of the processing on the client, rather than on the Datacap Web Client server.

AVerify client can manage processing tasks such as generating screens, loading data, and saving data. Therefore, AVerify is a good option if you are processing high volumes by using multiple web clients that are attached to the same web server. The server is started only for validations and for saving the XML files. Aside from this feature, VeriFine is usually a better choice for web verification. The following table describes the user interface (UI) controls (and the corresponding tooltip, if accessible) that are available in AVerify.

Table 1. AVerify UI Controls and Descriptions

UI Control	Description
Next LC	Highlights the next low confidence character on the current page.
◀ (left arrow, Previous page tooltip)	Displays the previous page in the batch.
Exclamation point (!) on an arrow that points left (Previous problem tooltip).	Submits the current page and displays the previous problem page.

UI Control	Description
Two vertical bars (Put batch on hold tooltip)	Puts the batch on hold.
Exclamation point (!) on an arrow that points right (Previous problem tooltip).	Submits the current page and displays the next problem page.
► (Right arrow, Next page tooltip)	Displays the next page in the batch.
Hold	Puts the batch on hold.
Submit	Submits the current page and displays the next problem page.
?	Runs the Verify task profile (validation rules).
Show snippet	Displays a window with an enlarged view of the current field.
Plus sign (+) on a magnifying glass (Zoom in tooltip)	Enlarges the page image view.
Minus sign (-) on magnifying glass (Zoom out icon)	Reduces the size of the page image view.
Magnifying glass on page (Zoom quarter tooltip)	Displays one quarter of the page.
Letter T on page graphic (Show words tooltip)	Outlines all of the words on the page.
Lines on a page graphic (Show lines tooltip)	Outlines all lines on the page image.
Letter T on page (Show fields tooltip)	Outlines all recognition fields on the page image.
Override check box	Select to override a validation failure.
Batch view...	Displays the data from the runtime batch hierarchy.

To use the AVerify client, select `averify.aspx` as the value for Program in the Verify task's setup dialog in the Datacap Web Client.

Creating and implementing custom (static) panels is covered in other topics.

AVerify uses the same page and field status settings and Rulerunner settings as the VeriFine client. For more information, see the *Configuring the page and field status settings* section in the [Configuring the VeriFine client](#) topic.

- [Creating and using custom \(static\) panels](#)
AVerify generates a default verification panel for each page type automatically. It maps each of the fields in the document hierarchy into a two-column table.

Parent topic: [Datacap Web Client and remote scanning](#)

Creating and using custom (static) panels

AVerify generates a default verification panel for each page type automatically. It maps each of the fields in the document hierarchy into a two-column table.

Procedure

To create and use custom static panels:

1. Export the default panel layout.
2. Customize the panel layout.
3. Specify the panel layout in the task's Setup dialog in the Datacap Web Client.

If you specify that AVerify is to use static panels but do not define a static panel for each page type, you receive a runtime error. AVerify is unable to display a verification panel.

Therefore, if you choose to use static panels, you must define a static panel for each page type.

- [Exporting the default panel layout](#)
A custom panel is referred to as a static panel. To create a static panel, you use Datacap Web Client Custom Layout Generator.
- [Customizing the panel layout](#)
The HTML file that AVerify exports for each page type defines each of the fields within a standard two-column table layout. Each cell represents one field and contains a label, an image snippet, and an edit control.
- [Specifying the custom panels to use in a task](#)
You control the use of custom (static) panels through the task setup in the Datacap Web Client.

Parent topic: [Verification by using the AVerify web client](#)

Exporting the default panel layout

A custom panel is referred to as a static panel. To create a static panel, you use Datacap Web Client Custom Layout Generator.

Procedure

To export the default panel layout:

1. Open the Datacap Web Client Custom Layout Generator (<http://<server>/tmweb.net/task/support/dragresize.htm>).
2. Browse to and select an .ascx file that corresponds to the panel layout that you want to export.
3. Move or resize any labels or fields, as wanted.
4. Click Save Panel.

The .ascx file is in one of these folders, depending on the version of Windows that you are using:

- C:\ (the root of the C: drive)
 - C:\Users*< username >*\AppData\Local\VirtualStore
5. Rename the file to match the page type. For example, you might name the file Rental_Agreement.aspx.
 6. Move the .ascx file into the C:\Datacap\tmweb.net\Task folder.
 7. Specify the custom panel in the task's setup.xml file.
 - a. Start the Datacap Web Client and log in the TravelDocs application with the `admin` account.
 - b. Click the Administrator tab, and then click Workflow.
 - c. Select the Verify task, and click Setup.
 - d. In the Verify.set.xml - Webpage dialog window, scroll to the Custom web panels section and click the Use custom web panels check box.
 - e. Enter a page type and the corresponding .ascx file name for the default panel that you want to export.

- f. Click Save and close the window.
 - Tip: To revert to the previous panel, clear the Use custom web panels check box and click Save.
8. Prepare a batch for verification.
9. Open the batch in AVerify and display at least one instance of each page type. For each page you open, Datacap creates an HTML file with the layout of the standard two-column verification panel for that page. The files are named *static page_type>.htm*, for example, *staticRental_Agreement.htm*. The files are saved in one of the following locations:
 - o C:\ (the root of the C: drive)
 - o C:\Users*<username>*\AppData\Local\VirtualStore

Parent topic: [Creating and using custom \(static\) panels](#)

Customizing the panel layout

The HTML file that AVerify exports for each page type defines each of the fields within a standard two-column table layout. Each cell represents one field and contains a label, an image snippet, and an edit control.

About this task

The HTML within the image snippet and edit control cells contains code that cannot be modified.

Image snippet	Edit control
<pre><OBJECT id=dcim_Pickup_Date title=Pickup_Date tabIndex=-1 codeBase="dcim.cab" classid=clsid:BA893287-8932-11D3-A0DB- 58B204C16365 width=191 height=30> ... </OBJECT>
</pre>	<pre><OBJECT onBlur=reSetValue(me) onkeydown=hkPress() id=txt_Pickup_Date language=vbscript class=AFlatEdit onfocus=Remember(me) title=Pickup_Date name=dcredit codeBase="dcim.cab" classid=clsid:D20D94B4-B85C-466C-B29B- 19B2ADAF60EC height=23> ... </OBJECT></TD></pre>

However, you can change the labels, rearrange the cells, remove the image snippets, and so on.

Important: When using application such as APT or TravelDoc, changing the zoom level of your Windows desktop to 150% may cause the layout of Invoice Number and Invoice Total fields, including the text labels, on the Static panel to be misaligned.

Procedure

To customize the default panel layout:

1. Open the HTML file in an HTML editor.
2. Make the required modifications to the labels, layout, and so on.
3. Save the file.

Parent topic: [Creating and using custom \(static\) panels](#)

Specifying the custom panels to use in a task

You control the use of custom (static) panels through the task setup in the Datacap Web Client.

Procedure

1. Open the Datacap Web Client and log on to the TravelDocs application.
2. Click the Administrator tab, and click Workflow.

3. Expand Web Job, and click the Verify task.
4. In the Selected task details pane, click Setup.
5. In the Verify.set.xml - Webpage dialog window, scroll to the Custom web panels section and click the Use custom web panels check box.
6. Under Bind page to ascx panel:
 - a. In the Panel for field, replace `Page_Type` by entering `Rental_Agreement` and replace `panel.htm` by entering `staticRental_Agreement.htm`.
 - b. Click the + sign to add a second Panel for field.
 - c. In the second Panel for field, replace `Key2` by entering `Optional_Insurance` and replace `Value2` by entering `staticOptional_Insurance.htm`.
7. Click Save.
8. Open a batch in AVerify and confirm that AVerify is using the new custom panels.
9. Optional: To revert to the default panels at any time:
 - a. Repeat steps 1 - 5 and click Use custom web panels to clear the check box.
 - b. Under Bind page to ascx panel:
 - Click the - sign to remove `Optional_Insurance`.
 - Replace `Rental_Agreement` by entering `Page_Type`.
 - Replace `staticRental_Agreement.htm` by entering `panel.htm`.
 - c. Click Save.

Parent topic: [Creating and using custom \(static\) panels](#)

Verification by using the ImgEnter web client

The ImgEnter (`imgenter.aspx`) web client is different from the other web verification clients in that you enter data through the page image view.

ImgEnter displays a gray border around each data field. When you click within a field, the web client displays a data entry edit field. The data entry field displays the recognized data that you can change. Fields with low confidence characters are displayed in yellow and fields with validation errors are displayed in red. To use the ImgEnter client, select `ImgEnter.aspx` as the Value for Program key in the Verify task's details pane in the Datacap Web Client. Click the Administrator tab, click Workflow, and then select Verify in the Web Job.

ImgEnter uses the same page and field status settings and RRS settings as the VeriFine client. For more information, see the *Configuring the page and field status setting* section in the [Configuring the VeriFine client](#) topic.

Parent topic: [Datacap Web Client and remote scanning](#)

Manual page identification and batch restructuring with ProtoId

You use the ProtoId web client (`ProtoId.aspx`) to do manual page identification. You can use the list beneath each thumbnail to change the current page type. Additionally, the small toolbar above each thumbnail image provides batch restructuring functions.

The following table lists the batch restructuring controls.

Control	Description	Control	Description
Plus sign (+)	Enlarges all thumbnails	Left arrow (<)	Moves the page to the left (ALT+U)
Minus sign (-)	Shrinks all thumbnails	Check mark (✓)	Indicates that the page was copied to the clipboard

Control	Description	Control	Description
keys	Displays hot key list	Copy	Copies the page to the clipboard (CTRL+C)
Question mark (?)	Runs document integrity check	Paste	Inserts the page from the clipboard* (CTRL+V)
		Curved arrow	Rotates the page thumbnail by 90 degrees (CTRL+G)
		right arrow (>)	Moves the page to the right (ALT+N)

When you copy a page, Datacap adds a page-level variable to the runtime hierarchy with the ID of the source page. For example, if you copy page 1, Datacap adds `<V n="Copy">TM000001</V>` to the cloned version. Tip: You can move between thumbnails by using TAB/SHIFT+TAB. You can display a full page image by clicking the thumbnail or pressing ENTER. For more information, see [ProtoID web client configuration](#).

- [ProtoID web client configuration](#)
You can insert new pages, modify the available page types, disable document integrity checks, and run rules with the ProtoID web client.

Parent topic: [Datacap Web Client and remote scanning](#)

ProtoID web client configuration

You can insert new pages, modify the available page types, disable document integrity checks, and run rules with the ProtoID web client.

To use the ProtoId client:

1. Start the Datacap Web Client, click the Administrator tab, and click Workflow.
2. Create or select a task for which you want to use the ProtoID client.
3. Under the Parameters heading in the Selected task details section, select ProtoID.aspx as the Value for the Program.
4. Click Save.

Inserting pages

The CTRL+M hot key inserts a new page before the selected page. To specify the type for the new page:

1. In the Datacap Web Client, click the Administrator tab, and click Workflow.
2. Select the task that uses ProtoID.aspx.
3. Click Setup in the Selected task details section.
4. In the `task.set.xml` Webpage Dialog, scroll down to the Page ID section and enter the name of an existing page type in the Insert type: field.

For example, if you enter `Separator_Page` here, and later during subsequent processing you press CTRL+M, ProtoId inserts a page of type `Separator_Page` and assigns the image file `blank.tif`. If you specify an invalid type, ProtoId assigns the type of page that is selected when you press CTRL+M.

5. Click Save.

Controlling the list of available page types

By default, ProtoId lists all available page types in the drop-down list below each thumbnail image.

If you want to limit the available page types or display aliases, create a dictionary of available page types in the application's document hierarchy XML file. The dictionary must have the name PageNames, for example:

```
<DICTIONARY n="PageNames">
  <W v="Page_Type_1">Page_Type_1</W>
  <W v="Page_Type_2">Page_Type_2</W>
  <W v="Page_Type_3">Page_Type_3</W>
</DICTIONARY>
```

Disabling document integrity checking

By default, ProtoId checks document integrity automatically when you click Done. You cannot complete the batch if there are integrity problems.

To disable automatic document integrity checking:

1. Start the Datacap Web Client, click the Administrator tab, and click Workflow.
2. Select the task that uses ProtoID.aspx.
3. Click Setup in the Selected task details section.
4. In the *task.set.xml* Webpage Dialog, scroll down to the Page ID section and clear the Document Integrity check box.
5. Click Save.

Running rules from ProtoID

ProtoId uses the same RRS settings as the VeriFine verification client (see [Verification by using the VeriFine web client](#)). The specified task profile runs immediately before document integrity checking.

Using "super variables"

The ALT+S hot key assigns a super variable to the selected page in the runtime batch hierarchy, for example:

```
<P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev039_TM00
0001">
  <V n="Super">Three</V>  <!-- Super variable assigned a value of "Three"
  <V n="STATUS">0</V>
  <V n="TYPE">Rental_Agreement</V>
etc.
```

The value is displayed above the page thumbnail image.

To specify the available super variable values, add the values to the Special Variable values field in the task's Setup dialog in the Datacap Web Client.

For example:

Special Variable values: One, Two, Three

Each time that you press ALT+S, ProtoId assigns the next value in the series. In this example, the first time you press ALT+S, it assigns a super variable value of One. The second time, it assigns Two; the third time, it assigns Three; the fourth time, it removes the Super variable.

You can use super variables to perform additional integrity checking by specifying a page type with each value, for example:

```
SuperVars=One|Rental_Agreement,Two|Air_Ticket,Three|Room_Receipt
```

ProtoId uses these `<value>|<page_type>` combinations during document integrity checking. In this example:

- Integrity checking fails if you assign `Super=One` to a page that is not of type `Rental_Agreement`.
- Integrity checking fails if you assign `Super=Two` to a page that is not of type `Air_Ticket`.
- Integrity checking fails if you assign `Super=Three` to a page that is not of type `Room_Receipt`.

In addition, you can use the tilde (~) character to indicate values that are not valid for a specific type, for example:

```
[PageID]
SuperVars=One|~Rental_Agreement
```

In this example, integrity checking fails if you assign `Super=One` to a page that is of type `Rental_Agreement`.

Parent topic: [Manual page identification and batch restructuring with ProtoId](#)

Administering an application

You use the Administrator tab in the Datacap Web Client for all administration tasks. You use this tab to configure your application from any machine on the network.

About this task

The Administrator tab gives you access to a setup window where you can configure task settings.

Procedure

To configure a task:

1. Start the Datacap Web Client and click the Administrator tab.
2. On the Workflow page, expand the job that contains the task that you want to configure.
3. Select the task that you want to configure.
4. In the Selected task details pane, you can:
 - a. Specify the Mode, such as Batch Creation or Router), and any queuing or storing options, by selecting a value from the corresponding menu.
 - b. Specify the program that the task uses, such as Rulerunner, Datacap Desktop, or one of several .aspx web pages.
 - c. Click Setup to open the task.set.xml - Webpage dialog window where you can select or configure more settings.
Important: The options that are available vary, depending on the program that you specify for the task. You must click Save in the Webpage dialog to save the additional configuration settings.
5. Click Apply in the Selected task details pane.

Parent topic: [Datacap Web Client and remote scanning](#)

Job monitoring

You can use Monitor tab in the Datacap Web Client to monitor the status of the job queue.

The labels at the top of the Monitor tab are active.

- The Items per page label controls how many jobs are displayed.
- The Delete batches label deletes all displayed batches. Use the Batch, Job, Task, or Status fields to control which jobs are displayed, or use the Filter button.

Tip: To delete an individual batch, click the batch number and then click Delete.

- The Filter label provides finer control over which jobs are displayed
- The Refresh label refreshes the job list (or set the rate to refresh automatically)
- The Default label returns to the default view (all jobs)

You can also use the web Monitor to perform the following tasks:

- Click the QID to run a batch
- Click the batch number to view the batch details, change the batch status, and optionally delete the batch.

Parent topic: [Datacap Web Client and remote scanning](#)

TravelDocs: Scanning from Datacap Web Client

You can create, configure, and run tasks that are related to scanning batches remotely by using Datacap Web Client.

- [Creating a remote scan task](#)
You are going to need a TWAIN scanner to create a remote scan task. Before you proceed, make sure that the scanner is connected and functioning.
- [Configuring the remote scanning client](#)
You confirm that the remote scanning client is configured to use the two-step scan-upload process.
- [Configuring the Upload task](#)
The default application framework includes an Upload task, but you need to create a shortcut.
- [Scanning and uploading a batch](#)
After you create and configure the scan and upload tasks, you can run them on your batches.
- [Creating the web Job CreateDocs task](#)
When you created the CreateDocs task, you created it for Main Jobs only. You must also create the task for web Jobs.
- [Configuring Rulerunner to run web jobs](#)
You can configure Rulerunner to run your web jobs remotely.
- [Modifying the Verify shortcut](#)
The default Verify shortcut is only configured for Main Jobs, not web jobs. Before you can open the batch for verification, you must modify the shortcut.
- [Opening the batch for verification](#)
You can verify the batch on the Datacap Web Client Operations tab.

Parent topic: [Datacap Web Client and remote scanning](#)

Creating a remote scan task

You are going to need a TWAIN scanner to create a remote scan task. Before you proceed, make sure that the scanner is connected and functioning.

About this task

Attention: If you want to run Datacap Web Client from a different machine, run the WebClientConfig utility on that machine to configure the required Internet Explorer security settings. You also need to add the Datacap Web Client (tmweb.net) server as a trusted site.

Procedure

To create a remote scan task:

1. Start Internet Explorer and open the Datacap Web Client home page:
 - o If the server is running on the same machine: <http://localhost/tmweb.net>
 - o If the server is running on a different machine: http://tmweb_server//tmweb.net
2. Log in to the TravelDocs application and click the Administrator tab.
3. Click Workflow, expand TravelDocs, and then expand Web Job.
4. Select the iVscan task and click Remove. Then, click OK.
5. Select Web Job and click New.
6. Enter the task details as follows:

Name: Scan	Description: Scan from Datacap Web Client	Mode: Batch Creation
Queue by: None	Store: None	Program: Scancl.aspx

7. Click Apply.
8. Select the new Scan task and click the Up Arrow to move the task to the top of the web Job task list.
Attention: If the new task disappears, expand any job with conditions and you can see the new task again.
9. Click Shortcuts and click New.
10. Enter the shortcut details as follows:

Name: RScan	Description: Remote Scanning
Mode: Auto	

11. Scroll down and under Web Job, check Scan. Then, click Save.
Important: If the web page stops responding, open the Datacap Server Manager, then stop and restart the server.

Parent topic: [TravelDocs: Scanning from Datacap Web Client](#)

Related information:

[Add tmweb.net address as a trusted site](#)

Configuring the remote scanning client

You confirm that the remote scanning client is configured to use the two-step scan-upload process.

Procedure

To configure the remote scanning client:

1. In the Datacap Web Client, click the Administrator tab, and then click Workflow.
2. Expand Web Job and then select Scan.
3. In the Selected task details section, click Setup.
4. In the setup window (Scan.set.xml - Webpage Dialog):
 - a. Scroll down to the Scan section.
 - b. Enter `c:\datacap\scan` for the Scan into directory field.
 - c. Confirm that the Local processing option is not selected Datacap scans the images to a local folder on the web client and then uses the Upload task to upload the images to the application batches folder.
 - d. Click Save.

Parent topic: [TravelDocs: Scanning from Datacap Web Client](#)

Configuring the Upload task

The default application framework includes an Upload task, but you need to create a shortcut.

Procedure

To configure the Upload task:

1. Start the Datacap Web Client, and log in to the TravelDocs application with the `Admin` credentials.
2. Click the Administrator tab and then click Shortcuts.
3. Click New.
4. Enter the shortcut details as follows:

Option	Description
Name:	Upload
Description:	Upload from Datacap Web Client
Mode:	Auto

5. Scroll to the Web Job option and select Upload. Then, click Save. If the web page is not responding after you save the shortcut, open the Datacap Server Manager window, and stop and restart the server.

Parent topic: [TravelDocs: Scanning from Datacap Web Client](#)

Scanning and uploading a batch

After you create and configure the scan and upload tasks, you can run them on your batches.

Procedure

To scan and upload a batch:

1. Print the file `ScanPage.pdf` that is included in the Sample Documents download.
2. In the Datacap Web Client, click the Operations tab and confirm that the two shortcuts that you created are present. If you do not see the shortcuts, click Logout and then log back on.
3. Click the RScan shortcut. Datacap Web Client displays the remote scanning page (`vscancl.aspx`).
4. Open a second web browser and open `tmweb.net`. Then, click the Monitor tab and set the Refresh rate to 10 sec. Arrange the web browser windows so that you can see them both.

The Monitor tab indicates that Datacap created the batch folder and created a new entry in the job queue.

5. In the remote scanning page, confirm that your scanner is selected and that the page is in the scanner's feeder or on the flatbed. Then, click Scan.
Attention: If your scanner does not have a feeder, you might see information messages about unsupported features. Click OK to continue.
6. When the scan completes and the page thumbnail is displayed, click Done. Then, click OK and Stop. In the Monitor tab, you can see the job as pending for the Upload task. (You might need to wait a moment.)
7. On the Operations tab, click the Upload shortcut and wait for the file to upload. When it completes, click OK. You can see the job as pending for the PageID task.
Attention: Rulerunner does not process the batch automatically because it is not configured to process web jobs. This processing is done later.
8. Open the most recent batches folder. (If you are using two machines, this folder is on the Datacap Server machine). You can see the image file and the two runtime batch files:
 - o `scan.xml` is the file that is generated by the RScan task
 - o `upload.xml` is the file that is generated by the Upload task

Creating the web Job CreateDocs task

When you created the CreateDocs task, you created it for Main Jobs only. You must also create the task for web Jobs.

Procedure

1. In Datacap web, click the Administrator tab.
2. On the Workflow subtab, expand the TravelDocs workflow and expand Web Job.
3. Select the Web Job node and click New to create a new task.
4. Enter the task details as follows:

Option	Description
Name:	CreateDocs
Description:	Create documents
Mode:	Normal
Queue by:	None
Store:	None

Attention: Since a task with the same name exists in the Main Job workflow, the fields auto-populate after you enter the name.

5. Click Apply.
6. Select the new CreateDocs task and click Up Arrow to move the task between PageID and Profiler.
Tip: If the new task disappears, expand any job with conditions to see the new task again.

Configuring Rulerunner to run web jobs

You can configure Rulerunner to run your web jobs remotely.

Procedure

To configure Rulerunner to run web jobs:

1. In the Start menu click IBM Datacap Services > Rulerunner Manager to open the Datacap Rulerunner Manager window.
2. On the Rulerunner tab, click Start to start the Datacap Rulerunner Service.
3. Click the Rulerunner Login tab and click Connect.
4. Click the Workflow: Job: Task tab and select TravelDocs in the left pane.
5. Under TravelDocs > Web Job, select PageID, CreateDocs, Profiler, and Export.
6. Drag Web Job to <thread0>.

<thread0> must be like this example:

- o <thread0>
- o TravelDocs
 - tms
 - <dbs>
 - Main Job

- PageID
 - Profiler
 - Export
 - CreateDocs
 - Web Job
 - PageID
 - CreateDocs
 - Profiler
 - Export
7. Click Save.
 8. Click the Rulerunner Login tab and click Disconnect.
 9. Click the Rulerunner tab and click Start.
 10. Open the Datacap Web Client and click the Monitor tab to watch Rulerunner process the web job through the PageID, CreateDocs, and Profiler tasks. Rulerunner stops when the web job is pending for the Verify task.

Parent topic: [TravelDocs: Scanning from Datacap Web Client](#)

Modifying the Verify shortcut

The default Verify shortcut is only configured for Main Jobs, not web jobs. Before you can open the batch for verification, you must modify the shortcut.

Procedure

1. In the Datacap Web Client, click the Administrator tab and then click Shortcuts.
2. Select the Verify shortcut.
3. Under the Permissions section in the Selected shortcut details pane, scroll to Web Job and click the Verify check box. Then, click Save.

Parent topic: [TravelDocs: Scanning from Datacap Web Client](#)

Opening the batch for verification

You can verify the batch on the Datacap Web Client Operations tab.

Procedure

1. Click the Datacap Web Client Operations tab, and then click Verify
Important: If the `All documents are complete` message is displayed, click Cancel so that you can review the page.
2. Scroll the second panel to view the check box options and the listbox control. You can see that the check box outlines are clipped, but the recognition engine is still able to identify the selected Fuel Service option.
Tip: To fix the clipped check box outlines, you might need to modify the image-processing settings based on a scanned page. Due to scanner differences, your scanned page might be different from the example that is described here. If the Fuel Service option is not selected, select Fuel Service before proceeding.
3. Click Submit and OK to finish the batch. Then, click OK and Stop.
4. Check the Datacap Web Client Monitor tab and observe Rulerunner complete the Export task.

Parent topic: [TravelDocs: Scanning from Datacap Web Client](#)

TravelDocs: Using AIndex for manual page identification and registration

You must be running Datacap 8.0.1 Fix Pack 1 or later to complete this section.

- [Making a copy of the application](#)
When you finish reviewing AIndex, you need to roll back the changes to the application. You can use the Datacap Studio Application Wizard to make a copy of the application and then work on the copy.
- [Updating the application](#)
You configured the PageID ruleset to use multiple page identification techniques that are implemented in a cascade fashion. To demonstrate manual page identification by using AIndex, you are going to remove the text matching and pattern matching functions and send any batch with unidentified pages to AIndex.
- [Updating ManualPageID](#)
You must configure the ManualPageID task to use aindex.aspx. You also must add the required page and field status settings and set the template path to point to the fingerprint folder of the application. Additionally, you must create a rule that runs if a user changes an existing page status.
- [Creating the ManualIDValidate rule](#)
AIndex assigns the failed (DOF) page status if you change an existing page type. You cannot complete the batch if any page has a failed status, so you need a rule to change the status to done.
- [Running a batch through the workflow](#)
Because Rulerunner is not configured to run the TravelDocs2 application, you need to use Datacap Web Client to run the batch through the background tasks.
- [Testing the ManualIDValidate rule](#)
Run another batch through the workflow but this time change the page status on one of the pages to start the ManualIDValidate profile.

Parent topic: [Datacap Web Client and remote scanning](#)

Making a copy of the application

When you finish reviewing AIndex, you need to roll back the changes to the application. You can use the Datacap Studio Application Wizard to make a copy of the application and then work on the copy.

Procedure

1. Start Datacap Studio but do not connect to any application. Instead, click Close in the Application window.
2. In the Datacap Studio window, click the Datacap Application Wizard button at the upper right.
3. In the Datacap Application Wizard dialog, click Next.
4. Select Copy an existing RRS application and click Next.
5. In the top field, select the TravelDocs application.
6. Leave the default values for the Root folder and Datacap Web Client folder fields.
7. Select Rename copy and enter TravelDocs2 in the New Name field.
8. Click Next and then click Finish. Then, wait for the copy to complete and click Close.

Parent topic: [TravelDocs: Using AIndex for manual page identification and registration](#)

Updating the application

You configured the PageID ruleset to use multiple page identification techniques that are implemented in a cascade fashion. To demonstrate manual page identification by using AIndex, you are going to remove the text matching and pattern matching functions and send any batch with unidentified pages to AIndex.

However, if you are using AIndex for manual page identification you must run the manual page identification task after you create a structured batch. You need to move the branching function out of the PageID task and into the CreateDocs task. You also need to make sure that each unidentified page is a separate document, which requires an update to the document hierarchy.

Updating the PageID ruleset

1. In the Datacap Studio window, click Connection Wizard at the upper right.
2. Select the TravelDocs2 application, click Next, enter the `admin` password, and click Finish.
3. Select the PageID ruleset and click Lock/Unlock ruleset for editing.
4. Expand the PageID ruleset and the PageID rule.
5. Remove the Identify using Text Match function and the Identify using Pattern Match function.
6. Expand the Identify Manually function and remove the `Task_NumberOfSplits` action and the `Task_RaiseCondition` action.
7. Change the parameters on the `rrSet` action as shown in the following table.

Library	Action	Parameter
rrunner	rrSet	<code>varSource = Yes</code> <code>varTarget = @B.ManualID</code>

8. Click Save. Then, click Lock/Unlock ruleset and choose Publish ruleset. The PageID rule should look like the following example.

Updating the Document Integrity ruleset

The CreateDocs task profile includes two rulesets: CreateDocs and Document Integrity. Because you need to branch to AIndex after you create the structure batch, you add the branching function to the Document Integrity ruleset.

Previously, you used the Document Integrity ruleset to raise a branch condition if the batch had structural problems that required manual fixup. You are going to modify this ruleset to raise the branch condition if the batch contains any pages that require manual identification.

1. Select the Document Integrity ruleset and click Lock/Unlock ruleset for editing.
2. Expand the Document Integrity ruleset, the Batch Document Integrity Check rule, and both functions.
3. Remove the `CheckAllIntegrity` action from the first function and replace it with the following action.

Library	Action	Parameter
rrunner	rrCompareNot	<code>object1 = @B.ManualID</code> <code>object2 = Yes</code>

Attention: This action returns `False` if the batch variable `ManualID` is `Yes`, which causes the rule to start the Batch Route To Fixup function. The PageID rule configures this variable to `Yes` if the batch includes pages that failed fingerprint matching.

4. Click Save. Then, click Lock/Unlock ruleset and choose Publish ruleset.

Updating the Document Hierarchy

1. In the document hierarchy pane, click Lock DCO for editing.
2. Expand the Document Hierarchy so that you can see the Other page and the Air Ticket page.

3. Right-click the Other page and choose Manage variables. Then, set the Order value to -1 and click Done. Attention: Setting the Order value to -1 causes the CreateDocs action to create a new document for each page of type Other.
4. Expand the Air_Ticket page.
5. Right-click the Vendor_Logo field and choose Manage variables.
6. Click New, type Required, and press Enter.
7. Set the value of the Required variable to 1 and click Done.
8. Click Save changes and then click Unlock DCO.

Changing the branch condition

1. Open the Datacap Web Client and click the Administrator tab.
2. Click Workflow tab, expand Main Job, expand CreateDocs, and select the Document Integrity Failed condition.
3. Click the down-arrow beside the Child Job field and choose ManalPageID Job.
4. Click Apply and then click Done.

Parent topic: [TravelDocs: Using AIndex for manual page identification and registration](#)

Updating ManualPageID

You must configure the ManualPageID task to use aindex.aspx. You also must add the required page and field status settings and set the template path to point to the fingerprint folder of the application. Additionally, you must create a rule that runs if a user changes an existing page status.

About this task

The page and field status settings for AIndex are the same as those settings for VeriFine (see [Configuring the VeriFine client](#)).

- [Ignored field statuses](#)
The Ignored Field Statuses field determines which fields to hide, meaning AIndex does not display any field that has one of the specified status values.
- [Done field statuses](#)
The Done Field Statuses field determines which fields to hide when the Problem fields only check box is selected. Because you are hiding all of the fields, it does not really matter which value you enter. However, you can enter 0 (the standard setting) for the Done Field Statuses field.
- [Done page statuses](#)
The Done Page Statuses field is used to determine when a batch is complete. When all of the pages in a batch have one of the specified values, you can complete the batch. Otherwise, you can put the batch on hold only.
- [Validation statuses](#)
The Validation Statuses field specifies three page status values.
- [Editing the ManualPageID settings](#)
The ManualPageID settings control how AIndex manages pages and fields with the specified status values.

Parent topic: [TravelDocs: Using AIndex for manual page identification and registration](#)

Ignored field statuses

The Ignored Field Statuses field determines which fields to hide, meaning AIndex does not display any field that has one of the specified status values.

Because you are using AIndex for manual page identification, there is no field data and it is not necessary to display a page of empty fields. Therefore, you can hide all of the fields by entering 0, -1 for the Ignored Field Statuses.

- 0 is the default status of each field when the page data file is created initially
- -1 is typically assigned to anchor fields so that they can be hidden

Parent topic: [Updating ManualPageID](#)

Done field statuses

The Done Field Statuses field determines which fields to hide when the Problem fields only check box is selected. Because you are hiding all of the fields, it does not really matter which value you enter. However, you can enter 0 (the standard setting) for the Done Field Statuses field.

Parent topic: [Updating ManualPageID](#)

Done page statuses

The Done Page Statuses field is used to determine when a batch is complete. When all of the pages in a batch have one of the specified values, you can complete the batch. Otherwise, you can put the batch on hold only.

By default, the VScan task assigns a status of 49 on all pages. In the PageID task, you assign a status of 1 to the batch if it includes pages that failed fingerprint matching.

When you manually assign a page type in AIndex, AIndex assigns the status that is specified as the first Validation status value (for example, 0, 2, 1, where Done status = 0; Override status = 2; Fail status = 1). You can assign a status of 0, and so you need to set a value of 0, 49 for the Done Page Statuses field. In this manner, the user can complete the batch only when all of the pages are identified.

Parent topic: [Updating ManualPageID](#)

Validation statuses

The Validation Statuses field specifies three page status values.

- The first value is the done status that is assigned when you set the page type for an unidentified page. When you set up the Done Page Status values, you assigned a value of 0.
- The second value is used for validation overrides. These values do not apply in this situation, but you can assign the standard override value, which is 2.
- The third value is assigned if you change an existing page type. However, it is also the failed status. AIndex does not complete the batch if a page has this status, regardless of what you put in the Done Page Status field. In most situations users do not change an existing page status, but they might mistakenly, or for some other reason, do so. You can manage this situation by using a rule (see [Creating the ManualIDValidate rule](#)). You can assign the value 99 initially and then use the rule to assign the done status.

To set the done status to 0, the override status to 2, and the failed status to 99, enter a value of 0, 2, 99 in the Validation Statuses field.

Parent topic: [Updating ManualPageID](#)

Editing the ManualPageID settings

The ManualPageID settings control how AIndex manages pages and fields with the specified status values.

Procedure

To edit the ManualPageID settings:

1. Open the Datacap Web Client, log on to the TravelDocs2 application, and click the Administrator tab.
2. On the Workflow page, expand ManualPageID Job, and select the ManualPageID task.
3. In the Selected task details pane, select aIndex.aspx in the Program field under the Parameters section.
4. Click Setup.
5. In the ManualPageID.set.xml -Webpage dialog window, enter these values for the corresponding fields:

Option	Description
Done Page Statuses	0,49
Validation Statuses	0,2,99
Done Field Statuses	0
Ignored Field Statuses	-1,0
Main Task Profile	ManualIDValidate
RuleRunner Service Log	3
Batch Log	1
RRS Type	LocalRRS
WRRS URL	http://127.0.0.1/RRS/
Template Folder	C:\Datacap\TravelDocs2\fingerprint

6. Click Save.

Parent topic: [Updating ManualPageID](#)

Creating the ManualIDValidate rule

AIndex assigns the failed (DOF) page status if you change an existing page type. You cannot complete the batch if any page has a failed status, so you need a rule to change the status to done.

About this task

The user must click Submit for each changed pageto run the rule. The rule does not do anything except return True, but this value is enough to cause AIndex to assign the done status to the page.

Procedure

1. In Datacap Studio, in the Rulesets pane, right-click TravelDocs2 and choose Add Ruleset.
2. Change the name of the new ruleset to ManualIDValidate and then select Function1.
3. Click the Actions library tab and expand the runner library.
4. Select the Status_Preserve_OFF action and click Add to function to add the action to Function1.
5. In the Rulesets pane, click Save. Then click Lock/Unlock ruleset and choose Publish ruleset.
6. Click the Task profiles tab and then click Lock/Unlock task profiles.

7. Click Add a new task profile, select Custom, type `ManualIDValidate`, and click OK.
8. Select the new `ManualIDValidate` profile, select the `ManualIDValidate` ruleset, and click Add ruleset to profile at the left of the Task Profiles pane.
9. Click Save and then click Lock task profiles.

Parent topic: [TravelDocs: Using AIndex for manual page identification and registration](#)

Running a batch through the workflow

Because Rulerunner is not configured to run the TravelDocs2 application, you need to use Datacap Web Client to run the batch through the background tasks.

Procedure

1. Using Windows Explorer, open `C:\Datacap\TravelDocs2\images`.
2. Delete the page `NewAirline.tif`.
3. Copy the file `OffsetAirTicket.tif` from the sample documents download into the `C:\Datacap\TravelDocs2\images` folder. (This file is the same file that is used earlier in the [Pattern Matching](#) section.)
Important: The `\images` folder must contain the files `Images_Page_01.tif` through `Images_Page_13.tif` and `OffsetAirTicket.tif`.
4. Open the Datacap Web Client and log on to the TravelDocs2 application.
5. In the Operations tab:
 - a. Click `VScan`, browse to the location of the image files, click open and click `Scan`. When the task completes, click `OK` and `Done`. Then, click `OK` and `Stop`.
 - b. Click `Upload`. When the task completes, click `OK` and `Stop`.
 - c. Click `CreateDocs` and wait for the task to complete. Then, click `Stop`.
6. Start Datacap Desktop, log in to TravelDocs, and select the `PageID` shortcut. When the task completes, click `OK`, and exit from Datacap Desktop.
7. In the Datacap Web Client Operations tab, click `CreateDocs` and wait for the task to complete. Then, click `Stop`.
8. In the Datacap Web Client, click the `Monitor` tab to open the `Job Monitor` page, which shows the batch that is pending for the `ManualPageID` task.
9. On the Operations tab, click `ManualPageID`. The AIndex web client displays the last page of the batch. Although the `PageID` task assigned the page type `Other` to this unidentified page, AIndex assigns the first page type, `Rental Agreement`, and forces you to change it because it has `STATUS=1`.
10. Use the menu at the top of the batch tree view pane to set the document type to `Flight` and the page type to `Air_Ticket`.
11. When you select the `Air_Ticket` page type, Datacap activates the `Anchors` button, which prompts you to set anchors for the page. Click `OK` to close the message box.
12. Click `Anchors`. Datacap prompts you to select a thumbnail image. Click `OK` to close the message box.
Attention: Datacap displays all fingerprints that have an anchor field with the same name as the anchor field defined for the selected page type. In this case, the `Vendor_Logo` field is defined only for the `Air_Ticket` page. So, only the air ticket fingerprint thumbnails are displayed.
13. Double-click the `Airline #2` fingerprint thumbnail (the second one). Then, use the mouse to align the red anchor object over the `Airline #2` vendor logo
14. Click `Done` to complete the batch. Then, click `OK` and `Stop`.
15. Since the batch is now pending for the `Profiler` task, Start Datacap Desktop, log in to TravelDocs, and select the `Profiler` shortcut. When the task completes, click `OK`, and exit from Datacap Desktop.
16. In the Datacap Web Client Operations tab, click `Verify` to open the batch in the AIndex web client for verification.
17. Complete the batch by clicking `Submit` for each page. On page `TM000004`, set the car type to `Other`. On pages `TM000006` and `TM000013`, select the yellow checkbox at the top of the page to override the

validation failures.

Tip: You might need to scroll till end of the data entry panel to see the Submit button. Also, you cannot select multiple options for the Options groups on the Rental_Agreement pages because the AIndex web client does not support the `MultiPunch` variable.

Parent topic: [TravelDocs: Using AIndex for manual page identification and registration](#)

Testing the ManualIDValidate rule

Run another batch through the workflow but this time change the page status on one of the pages to start the ManualIDValidate profile.

Procedure

1. Start and log in to the Datacap Web Client, and click VScan on the Operations tab.
2. In the VScan window, click Browse, go to C:\Datacap\TravelDocs\images. Select a file and click Open, then click Scan.
3. When the VScan task displays a message to indicate that the scanning is complete, click OK, click Done then click OK and Stop.
4. On the Datacap Web Client Operations tab, click Upload.
5. When the Upload task displays a message to indicate that the task is complete, click OK and then click Stop to return to the Operations tab.
6. Start Datacap Desktop, log in to TravelDocs, and select the PageID shortcut. When the task completes, click OK.
7. In the Datacap Web Client Operations tab, click the CreateDocs icon and wait for the task to complete. Then, click Stop.
8. On the Datacap Web Client Operations tab, click the ManualPageID shortcut.
9. For the last page, set the document type to Flight and the page type to Air_Ticket. Then, click Anchors to assign the matching fingerprint and register the page, as you did before.
10. In the batch tree view pane, scroll up to the page list and select the first Rental_Agreement page.
11. Use the dropdown list to change the document type to Hotel. Then, change it back to Car_Rental. AIndex sets the page status to failed (99).
Attention: If you want to confirm this status, click Hold to put the batch on hold and then open the file manualpageid.xml in the current batch folder. You see `<V n="STATUS">99</V>` under page TM000001. Then, reopen the batch by clicking the batch ID on the Datacap Web Monitor tab.
12. With the Car Rental #1 page displayed, click Submit. AIndex runs the validation rule and changes the page status to 0.
13. Click Done to complete the batch. Then, click OK and Stop.

Parent topic: [TravelDocs: Using AIndex for manual page identification and registration](#)

Filter batches by group in the Job Monitor (Datacap Web Client)

In the Datacap Web Client, you can filter batches by groups in the Job Monitor based on your ADSI, LDAP, or LLDAP group authentication.

The view of batches in Job Monitor can be restricted to batches that are assigned to a specific group. The application's rules determine the groups that can view the batch. Four steps must be completed to filter batches by group in the Job Monitor.

- You must set up your ADSI, LDAP, or LLDAP group authentication in the Datacap Server Manager. If LLDAP is used, the group-based authentication option must be used. For information on setting up LLDAP group authentication, see *LLDAP group authentication*
- A custom column must be defined in the tmBatch table in the Datacap Engine database. For information on defining a custom column, see *Creating a custom column in the Job Monitor*.
- The group names must be set up for filtering purposes by defining a custom Job Monitor filter in the Application Manager. The two methods for filtering are with exclusive groups and additive groups.
 - You can define a maximum of 31 additive groups. Each user can be member of multiple groups and each batch can be assigned multiple groups. The user must belong to all of the groups that are assigned to the batch to view that batch.
 - You can define up to 32,000 exclusive groups. Each user can be a member of only one exclusive group and a batch can be assigned to one filter group. You can create a supervisor group whose members can view all batches with exclusive filters.
- You must assign one or more groups to each batch to be filtered by adding a rule in Datacap Studio. When groups are assigned to a batch by this rule, the batch can be viewed in Job Monitor only by members of the assigned group.

The batches that you run by selecting Datacap Web Client > Operations > Run Shortcut can be limited by using additive group filters in the same way that those batches are limited in Job Monitor. You cannot use exclusive group filters to limit batches that are run when you select Run Shortcut.

- [Defining group names for filtering batches](#)
You can define group names in the Application Manager for filtering batches by group.
- [Assigning a group to a batch for filtering](#)
You can assign one or more groups to each batch to be filtered by defining a custom Job Monitor filter and adding a rule in Datacap Studio. When the groups are assigned to a batch, the batch can be viewed in Job Monitor only by members of the assigned groups.
- [Configuring the filter method](#)
The configuration in Application Manager is static and defines how the feature operates in the application. For setting up batch filtering, you must configure Application Manager settings `CustomJMFilter` and `CustomMultiGroup`.

Parent topic: [Datacap application development](#)

Related information:

[LLLDAP group authentication](#)

[Creating a custom column in the Job Monitor](#)

Defining group names for filtering batches

You can define group names in the Application Manager for filtering batches by group.

About this task

Add each ADSI, LDAP, or LLDAP filtering group to the Custom Values tab of the Application Manager.

You can define a maximum of 31 additive groups. Each user can be member of multiple groups. Each batch can be assigned multiple groups and the user must belong to all of the groups that are assigned to the batch.

You can define up to 32,000 exclusive groups. Each user can be a member of only one exclusive group and a batch can be assigned to one filter group. You can create a supervisor group whose members can view all batches with exclusive filters.

Procedure

1. Open the Application Manager and select the application that you want from the list of applications.
2. Select the Custom values tab.
3. In the General string values field, add each filter group Value name and Value.

The group Value Name consists of the word `Group` followed by a number.

- o For additive groups, the number is 0 - 31
- o For exclusive groups, the number is 1 - 32767.

The Value of each group is the name of the ADSI, LDAP, or LLDAP user group to associate with the internal group number. The group name value is not case-sensitive but it must include the domain, if any, such as `NYOffice.SomeDomain`.

Parent topic: [Filter batches by group in the Job Monitor \(Datacap Web Client\)](#)

Assigning a group to a batch for filtering

You can assign one or more groups to each batch to be filtered by defining a custom Job Monitor filter and adding a rule in Datacap Studio. When the groups are assigned to a batch, the batch can be viewed in Job Monitor only by members of the assigned groups.

Procedure

Add a rule in Datacap Studio to assign one or more groups to each batch.

- You can assign groups immediately when the batch is scanned. Add the rule to a validation task profile for an interactive start batch panel that is based on the scan operator, job type, or other metadata that is defined in the start batch panel.
- You can assign groups later in the workflow as part of background processing rules that are based on any metadata in the batch.

Your rule must call a custom action to set the custom Job Monitor column named `pb_allowed`. Set the value to group number for exclusive groups or the sum of additive group values.

Parent topic: [Filter batches by group in the Job Monitor \(Datacap Web Client\)](#)

Configuring the filter method

The configuration in Application Manager is static and defines how the feature operates in the application. For setting up batch filtering, you must configure Application Manager settings `CustomJMFilter` and `CustomMultiGroup`.

Procedure

Define the custom Job Monitor filter by adding the `CustomJMFilter` and `CustomMultiGroup` values to the Application Manager.

1. Open the Application Manager and select the application that you want from the list of applications.
2. Select the Custom values tab.
3. In the General string values field, add the custom Job Monitor filter Value name, `CustomMultiGroup`. Set the value to 0 for exclusive group filtering or set the value to 1 for additive filters. The default value is 1.
4. In the General string values field, add the custom Job Monitor filter Value name, `CustomJMFilter`. The appropriate value depends on your database type and filter type. If the custom column in Job Monitor is

named, `pb_allowed`, define the additive or exclusive filter as follows.

- For additive filtering with Oracle, enter `bitand(CAST(pb_allowed As Integer), {0}) = CAST(pb_allowed As Integer)`
- For additive filtering with SQL Server, enter `CAST(pb_allowed AS INT) & {0} = CAST(pb_allowed AS INT)`
- For exclusive filtering with Oracle, enter `CAST(pb_allowed As Integer) = {0}`
- For exclusive filtering with SQL Server, enter `CAST(pb_allowed AS INT) = {0}`
- For exclusive filtering where Group1 is the supervisor group with SQL Server, enter `(CAST(pb_allowed AS INT) = {0}) OR 1 = {0}`
- For exclusive filtering with MS-Access, enter `CInt(pb_allowed) = {0}`

Parent topic: [Filter batches by group in the Job Monitor \(Datacap Web Client\)](#)

Fingerprint Management

Fingerprints are used both for page identification and for specifying recognition zones. The following topics review basic fingerprint functions, provide more details about the fingerprint database, and examine an alternative method for storing zone position information with fingerprint XML (FPXML) files. Later, you can update the TravelDocs application to use FPXML.

- [Review of basic fingerprint functionality](#)
In Datacap applications, fingerprints have two basic functions.
- [The Fingerprint database](#)
The information to manage the application fingerprint files is stored in the Fingerprint database of the application. By default, the Access database file (`<app_name>Fingerprint.mdb`) is stored in the root of the application folder.
- [Using fingerprint XML files](#)
A fingerprint typically consists of an entry in the fingerprint database, a TIFF image file, a CCO fingerprint file, and position information stored in the document hierarchy XML file.
- [TravelDocs: Updating auto fingerprinting to use FPXML](#)
You can update the AutoFingerprint ruleset to save new fingerprint position information in a separate fingerprint XML file. You can also update the Recognize Page rule to read the zone information from a fingerprint XML file, if one exists.

Parent topic: [Datacap application development](#)

Review of basic fingerprint functionality

In Datacap applications, fingerprints have two basic functions.

- You can use them during page identification to determine if the incoming page matches a known page:
- You can use them to identify the field positions for each variant of each known page type:
- [Create fingerprint files](#)
Datacap provides two methods for creating fingerprint files, image analysis and full page recognition.
- [Add fingerprints to the fingerprint library](#)
You can add fingerprints to the fingerprint library from the Datacap Studio Zones tab. Each time that you add a fingerprint, Datacap starts the *FingerprintAdd* rule set. In the fingerprint library, you can define the fingerprint generation method, such as image analysis or full page recognition.
- [Define field zones](#)
The position of each field zone is defined by coordinates (x1, y1, x2, y2) that specify the upper left and lower right corners of the zone relative to the upper left corner of the page.

Create fingerprint files

Datacap provides two methods for creating fingerprint files, image analysis and full page recognition.

- Image analysis: This method scans the page image to identify the composite blackness of different regions of the page. It does fast page identification, but requires that you perform recognition later.
- Full page recognition: This method performs optical character recognition to identify the locations of text within the page. The full page recognition method takes longer, especially with pages that include handwritten text. However, this method reduces time from subsequent workflow tasks because the full page recognition results are available for use.

For both methods, Datacap creates two files each time you generate a new fingerprint.

- A TIFF file with an image of the page.
- A CCO file with the fingerprint information.

If you are using full page recognition, Datacap creates a temporary XML file that is generated during recognition.

Attention: The method that you use for creating library fingerprints must be the same as the method you use to generate runtime fingerprints during page identification.

- An XML file (<fingerprint_id>c.xml) with the recognition results.

Parent topic: [Review of basic fingerprint functionality](#)

Add fingerprints to the fingerprint library

You can add fingerprints to the fingerprint library from the Datacap Studio Zones tab. Each time that you add a fingerprint, Datacap starts the *FingerprintAdd* rule set. In the fingerprint library, you can define the fingerprint generation method, such as image analysis or full page recognition.

You can add new fingerprints and define the recognition zones using actions. You can also create fingerprints for manually identified pages. (See [Creating the AutoFingerprint ruleset](#).)

Fingerprint files are stored in the application's fingerprint folder. The location of this folder is specified in the Datacap Application Manager and stored in the application configuration (.app) file.

In addition to saving the TIFF and CCO files, Datacap creates an entry for the new fingerprint in the fingerprint database. (See [The Fingerprint database](#)). The fingerprint database is also specified in the .app file.

Parent topic: [Review of basic fingerprint functionality](#)

Define field zones

The position of each field zone is defined by coordinates (x1, y1, x2, y2) that specify the upper left and lower right corners of the zone relative to the upper left corner of the page.

During recognition, Datacap uses the zone information to determine where the required information is on the page. If the runtime page image is not aligned precisely with the matched fingerprint, Datacap uses the calculated offset values to adjust the zone positions.

When you define the field zones in Datacap Studio, Datacap writes the position information into the document hierarchy. If you are defining the zones during verification, the `iloc_SetZones` action does the same. The document hierarchy XML file example shows the position of the `Pickup_Date` field for three different fingerprints:

```
<F type="Pickup_Date">
  <V n="ID">0</V>
  <V n="TYPE">Field</V>
  <V n="STATUS">0</V>
  <V n="POSITION">0,0,0,0</V>
  <V n="MIN_TYPES">0</V>
  <V n="MAX_TYPES">0</V>
  <V n="ReqConf">8</V>
  <V n="rules">&lt;in&gt;&lt;r id="1" rs="9" /></V>
  <V n="Pos556">183,402,535,463</V>
  <V n="Pos558">568,331,967,389</V>
  <V n="Pos560">1199,389,1600,448</V>
</F>
```

The last three entries in the XML file indicate the Zone position for fingerprints 556, 558, and 560.

It is also possible to store the field position information for each fingerprint in a separate file. For more information, see [Using fingerprint XML files](#). This is helpful if your application has many fingerprints.

Parent topic: [Review of basic fingerprint functionality](#)

The Fingerprint database

The information to manage the application fingerprint files is stored in the Fingerprint database of the application. By default, the Access database file (`<app_name>Fingerprint.mdb`) is stored in the root of the application folder.

The connection string for the Fingerprint database is defined through the Datacap Application Manager and stored in the application configuration file.

The fingerprint database includes three tables:

- **Host:** This table defines the name, host ID, and reference ID for each fingerprint class, for example:

Name	Host ID	Reference ID
<Global>	9	-1
Car_Rental	226	DC226
Flight	227	DC227
Hotel	228	DC228

Tip: You can see the host ID and reference ID by moving the mouse pointer over the class name in Datacap Studio.

- **PageType:** This table defines the name and ID for each page type, for example:

Page Type Name	Page Type ID
Other	1
Rental_Agreement	40
Optional_Insurance	41
Air_Ticket	42

Page Type Name	Page Type ID
Room_Receipt	43
Meals	44
Other_Charges	45

- Template: This table defines the ID, CCO file, TIFF file, host ID (class), and page type for each fingerprint, for example:

ID	CCO Path	Image Path	Host ID	Page Type
555	C:\Datacap\...\fingerprint\555.cco	C:\Datacap\...\fingerprint\555.tif	9	1
556	C:\Datacap\...\fingerprint\556.cco	C:\Datacap\...\fingerprint\556.tif	226	40
557	C:\Datacap\...\fingerprint\557.cco	C:\Datacap\...\fingerprint\557.tif	226	41
558	C:\Datacap\...\fingerprint\558.cco	C:\Datacap\...\fingerprint\558.tif	227	42
559	C:\Datacap\...\fingerprint\559.cco	C:\Datacap\...\fingerprint\559.tif	228	43

Parent topic: [Fingerprint Management](#)

Using fingerprint XML files

A fingerprint typically consists of an entry in the fingerprint database, a TIFF image file, a CCO fingerprint file, and position information stored in the document hierarchy XML file.

This setup works well when the number of fingerprints is small, but as the number of fingerprints increases the document hierarchy file gets bigger and the time to locate the position information increases.

- [The fingerprint XML file](#)
You can move the zone position information of a fingerprint out of the document hierarchy and into a separate fingerprint XML (FPXML) file.
- [Enable FPXML](#)
You can add fingerprints using the Datacap Studio Zones tab or by using actions from a verification panel.
- [Exporting existing position information from the document hierarchy](#)
The techniques that are described in the previous section work for fingerprints that are generated after you decided to use FPXML. For existing fingerprints, the zone position information is still in the document hierarchy.

Parent topic: [Fingerprint Management](#)

The fingerprint XML file

You can move the zone position information of a fingerprint out of the document hierarchy and into a separate fingerprint XML (FPXML) file.

Document hierarchy XML:

```
<F type="Pickup_Date">
  <V n="ID">0</V>
  <V n="TYPE">Field</V>
  <V n="STATUS">0</V>
  etc.
  <V n="Pos556">183,402,535,463</V>
```

```

                <V n="Pos558">568,331,967,389</V>
                <V n="Pos560">1199,389,1600,448</V>
            </F>
    <F type="Pickup_Location">
        <V n="ID">0</V>
        <V n="TYPE">Field</V>
        <V n="STATUS">0</V>
        etc.
        <V n="Pos556">180,528,532,589</V>
        <V n="Pos558">573,448,967,502</V>
    </F>

```

Fingerprint XML file:

```

<S>
  <P type="Rental_Agreement">
    <V n="HostID">226</V>
    <V n="HostName">Car_Rental</V>
    <F type="Pickup_Date">
      <V n="Postion">183,402,535,463</V>
    </F>
    <F type="Pickup_Location">
      </V>
      <V n="Postion">180,528,532,589</V>
    </F>
    etc.
  </P>

```

Although the total number of files is increased because each fingerprint now has a TIFF file, a CCO file, and an XML file, this approach has several benefits:

- The document hierarchy XML file remains small.
- Overall performance can increase.
- It eliminates possible contention if multiple users attempt to add fingerprints simultaneously.

Attention: These benefits apply to applications that use dynamic, action-based fingerprint generation, because Datacap Studio writes to the document hierarchy and the fingerprint XML file when FPXML is enabled.

Parent topic: [Using fingerprint XML files](#)

Enable FPXML

You can add fingerprints using the Datacap Studio Zones tab or by using actions from a verification panel.

- [Adding fingerprints using the Datacap Studio Zones tab](#)
By default, Datacap Studio writes the zone position information to the application's document hierarchy XML file, but can write to a fingerprint XML file.
- [Add fingerprints using actions](#)
The FPXML library includes actions for reading and writing zone information using fingerprint XML files.

Parent topic: [Using fingerprint XML files](#)

Adding fingerprints using the Datacap Studio Zones tab

By default, Datacap Studio writes the zone position information to the application's document hierarchy XML file, but can write to a fingerprint XML file.

About this task

To configure Datacap Studio to write zone position information to a fingerprint XML file, select the Enable FPXML option in the Application Manager.

After you enable FPXML and restart Datacap Studio, each time you add a fingerprint or modify an existing fingerprint's zones, Datacap Studio creates or updates the fingerprint XML file.

Important: Datacap Studio also adds the information to the document hierarchy XML file since it uses this information to display the zone outlines on the Image tab.

Parent topic: [Enable FPXML](#)

Add fingerprints using actions

The FPXML library includes actions for reading and writing zone information using fingerprint XML files.

Library	Action	Description
FPXML	SetDirectoryFPX	Sets the location for the fingerprint XML files.
FPXML	ReadZonesFPX	Loads the zone position information for the current fingerprint.
FPXML	WriteZonesFPX	Writes the position information for all fields on the current page.

For a demonstration of how to use these actions when you update the TravelDocs application, see the topic [Updating the AutoFingerprint ruleset](#).

Parent topic: [Enable FPXML](#)

Exporting existing position information from the document hierarchy

The techniques that are described in the previous section work for fingerprints that are generated after you decided to use FPXML. For existing fingerprints, the zone position information is still in the document hierarchy.

With the Fingerprint Maintenance Tool, you can export existing position information from the document hierarchy and into individual XML files. The Fingerprint Maintenance Tool is included as an APT-specific utility. However, it is possible to use the tool with other Datacap applications by following the instructions that are provided.

Important: If you move existing zone position information out of the document hierarchy and into FPXML files, you must update the actions that your application uses to read the zone information. The ReadZones action reads information from the document hierarchy, whereas the ReadZonesFPX action reads information from FPXML files.

- [Setting up the Fingerprint Maintenance Tool for your application](#)
You need to copy the Fingerprint Maintenance Tool files to your application directory and configure the tool for use with your application.
- [Exporting the position information](#)
This procedure removes the zone position information from the document hierarchy XML file (C:\Datacap*app_name*\dco_<i>app_name\<i>app_name.xml). Make a backup copy of this file before you export.

Parent topic: [Using fingerprint XML files](#)

Setting up the Fingerprint Maintenance Tool for your application

You need to copy the Fingerprint Maintenance Tool files to your application directory and configure the tool for use with your application.

Procedure

1. In the folder C:\Datacap\APT\dco_APT, locate the files Fingerprint Maintenance Tool.exe and Interop.DCAppleLib.dll.
2. Copy the two files from C:\Datacap\APT\dco_APT to C:\Datacap*<app_name>*\dco_*<app_name>*, where *<app_name>* is the name of your application)
3. Using a text editor, create a file that is called Settings.ini in the C:\Datacap*<app_name>*\dco_*<app_name>* folder and insert the following information (replace *<app_name>* with the name of your application):

```
[Database]
FingerprintDatabase=Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Datacap\<app_name>\<i><app_name>Fingerprint.mdb;Persist Security
Info=False
[Paths]
FingerprintDirectory=C:\Datacap\<app_name>\fingerprint
FingerprintBackupDirectory=C:\Datacap\<app_name>\Fingerprint Backup
SetupDCO=C:\Datacap\<app_name>\dco_<app_name>\<i><app_name>.xml
[FMT]
FilteredSummary=Select
Template.tp_TemplateID,Template.tp_DateAdded,Template.tp_HitCount,Template.tp_L
astHit,Host.hs_RefName FROM Template,Host WHERE host.hs_HostID =
Template.tp_HostID
```

Parent topic: [Exporting existing position information from the document hierarchy](#)

Exporting the position information

This procedure removes the zone position information from the document hierarchy XML file (C:\Datacap*<app_name>*\dco_*<app_name>*\<i><app_name>.xml). Make a backup copy of this file before you export.

Procedure

1. Double-click Fingerprint Maintenance Tool.exe.
2. Confirm that the information displayed at the top of the FMT window is correct.
3. Click the DCO to FPXML button and click OK to create the FPXML files.

Parent topic: [Exporting existing position information from the document hierarchy](#)

TravelDocs: Updating auto fingerprinting to use FPXML

You can update the AutoFingerprint ruleset to save new fingerprint position information in a separate fingerprint XML file. You can also update the Recognize Page rule to read the zone information from a fingerprint XML file, if one exists.

- [Updating the AutoFingerprint ruleset](#)
You can update the AutoFingerprint ruleset to save new fingerprint position information in a separate

fingerprint XML file.

- [Updating the Recognize Page rule](#)
You can update the Recognize Page rule to read the zone information from a fingerprint XML file.
- [Preparations for running a batch through the workflow](#)
Confirm that you have the correct files for this exercise before you run the batch through the workflow.
- [Running a batch through the workflow](#)
Now that you have completed updating auto fingerprinting to use FPXML, you can test it by running a batch through the workflow.

Parent topic: [Fingerprint Management](#)

Updating the AutoFingerprint ruleset

You can update the AutoFingerprint ruleset to save new fingerprint position information in a separate fingerprint XML file.

Procedure

To update the AutoFingerprint ruleset:

1. Start Datacap Studio and connect to the TravelDocs application.
2. On the Rulemanager tab, in the Rulesets pane, select the AutoFingerprint ruleset and click Lock/Unlock ruleset.
3. Expand the AutoFingerprint ruleset completely.
4. Right-click the ilocSetZones action and choose Remove.
5. Add the actions and parameters in the table below to the end of Function1.

Library	Action	Parameter
FPXML	SetDirectoryFPX	@APPPATH(fingerprint)
FPXML	WriteZonesFPX	@D.TYPE,@P.TYPE,@P.TYPE

Attention: The WriteZonesFPX action sets the fingerprint host name to the current document type, the fingerprint host ID to the current page type, and the fingerprint page type to the current page type.

6. Click Save. Then click Lock/Unlock ruleset and choose Publish Ruleset.

Parent topic: [TravelDocs: Updating auto fingerprinting to use FPXML](#)

Updating the Recognize Page rule

You can update the Recognize Page rule to read the zone information from a fingerprint XML file.

Procedure

1. On the Datacap Studio Rulemanager tab, in the Rulesets pane, select the Recognize ruleset and click Lock/Unlock ruleset.
2. Expand the Recognize ruleset and the Recognize Page rule.
3. Change the name of the function to Recognition: Fingerprint-Non-FPXML.
4. Remove the rrCompareNot action and replace it with the following action and parameters:

Library	Action	Parameter
rrunner	rrCompare	object1 = @P.MatchType object 2 = Fingerprint

5. Right-click the Recognize Page rule and choose Add Function.
6. Rename the new function to Recognition: Fingerprint-FPXML and use the Up Arrow button to move it to the beginning of the rule (before the other recognition function).
7. Add the actions and parameters below to the Recognition: Fingerprint - FPXML function.

Library	Action	Parameter
rrunner	rrCompare	object1 = @P.MatchType object 2 = Fingerprint
FPXML	SetDirectoryFPX	@APPPATH(fingerprint)
FPXML	ReadZonesFPX	
Recog_Shared	SnapCCOtoDCO	

8. Click Save. Then click Lock/Unlock ruleset and choose Publish Ruleset.
Attention: ReadZonesFPX fails if there is no matching FPXML file and Datacap runs the Non-FPXML function.

Parent topic: [TravelDocs: Updating auto fingerprinting to use FPXML](#)

Preparations for running a batch through the workflow

Confirm that you have the correct files for this exercise before you run the batch through the workflow.

Procedure

1. Start the Rulerunner Manager. If the Rulerunner service is running, click Stop and wait for the service to stop.
2. In Datacap Studio, click the Zones tab and then click the Refresh button.
3. Check to see if there is already a fingerprint for the Airline #4 ticket. If there is, select it and click the Remove selected button. The Airline #4 fingerprint, if you have one, is under its own Flight class.
4. In Windows Explorer, open C:\Datacap\TravelDocs\images and confirm that you have the files Images_Page_01.tif through Images_Page_13.tif and NewAirline.tif.

Parent topic: [TravelDocs: Updating auto fingerprinting to use FPXML](#)

Running a batch through the workflow

Now that you have completed updating auto fingerprinting to use FPXML, you can test it by running a batch through the workflow.

Procedure

1. Start Datacap Web Client and log in to the TravelDocs application.
2. On the Operations tab, click VScan, browse to the location of the image files, click open and click Scan. When the task completes, click OK and Done. Then, click OK and Stop to return to the Operations tab.
3. Click Upload. When the task completes, click OK and Stop to return to the Operations tab.
4. Start Datacap Desktop, log in to TravelDocs, and select the PageID shortcut. When the task completes, click OK and exit from Datacap Desktop.
5. In the Datacap Web Client Operations tab, click the ManualPageID shortcut and wait for the page images to load. Then, scroll to the bottom and set the page type for the last page to Air_Ticket.
6. Click Done and then, click OK and Stop.
7. In the Operations tab, click the CreateDocs shortcut. When it completes, click Stop.

8. Start Datacap Desktop, log in to TravelDocs, and select the Profiler shortcut. When the task completes, click OK and exit from Datacap Desktop.
9. In the Datacap Web Client, check the Job Monitor. You can see the result of the split, where the child job is pending for the Supervisor Verify task and the main job is pending for the Main Verify task.
10. On the In the Operations tab, click the Supervisor Verify shortcut to open the pending batch. The Airline #4 page is displayed.
11. Define the zone for each field as you did earlier (see [Running a batch through the workflow](#)).
12. Click Submit and then, click OK. In the background, Datacap runs the AutoFingerprint ruleset to create the new fingerprint and the fingerprint XML file. Then, click OK and Stop.
13. In Windows Explorer, open C:\Datacap\TravelDocs\fingerprint and locate the most recent fingerprint. You can see a .cco, a .tif, and a .xml file for the new fingerprint.
14. Open the fingerprint XML file in a text editor. Here is an example:

```

<S>
  <P type="Air_Ticket">
    <V n="HostID">Air_Ticket</V>
    <V n="HostName">Flight</V>
    <F type="Outbound_From">
      <V n="Position">649,488,1046,537</V>
    </F>
    <F type="Outbound_To">
      <V n="Position">1052,486,1477,545</V>
    </F>
    <F type="Outbound_Date">
      <V n="Position">182,484,386,541</V>
    </F>
    <F type="Return_From">
      <V n="Position">646,619,1023,674</V>
    </F>
    <F type="Return_To">
      <V n="Position">1053,621,1475,671</V>
    </F>
    <F type="Return_Date">
      <V n="Position">188,617,360,675</V>
    </F>
    <F type="Airfare">
      <V n="Position">1374,884,1513,928</V>
    </F>
    <F type="Taxes">
      <V n="Position">1391,936,1509,981</V>
    </F>
    <F type="Total_Cost">
      <V n="Position">1387,989,1522,1037</V>
    </F>
  </P>
</S>

```

Attention: If you run another batch through the workflow, it runs from end to end with no branching or splitting because the Airline #4 page is now in the fingerprint library. This time recognition task profile uses the zone position information from the fingerprint XML file.

Parent topic: [TravelDocs: Updating auto fingerprinting to use FPXML](#)

Configuring content classification for XML layout block parsing

Some XML configuration file changes might be needed for IBM® Content Classification to properly parse the text blocks sent to it by the RunDecisionPlanForBlocks action.

Procedure

To configure content classification for XML layout block parsing:

1. Open the file %ContentClassificationInstallDir%\Filters\docFilterManager.xml in an XML editor or text editor. For example, the file might be in the following directory: C:\IBM\ContentClassification\Filters.
2. Search for the string "XPATH-DATACAP".
3. If the string is not present in the file:
 - a. Make a backup copy of the docFilterManager.xml file.
 - b. Search for the XML element AllFilters.
 - c. Add the following Filter element to the AllFilters node:

```
<Filter>
  <FilterName>XPATH-DATACAP</FilterName>
  <Timeout>30</Timeout>
  <NvpMapping>
    <Property>
      <PropName>/P/Block/</PropName>
      <NvpName>Block</NvpName>
    </Property>
    <Property>
      <PropName>/P/Header/</PropName>
      <NvpName>Header</NvpName>
    </Property>
    <Property>
      <PropName>/P/Footer/</PropName>
      <NvpName>Footer</NvpName>
    </Property>
    <Property>
      <PropName>/P/Title/</PropName>
      <NvpName>Title</NvpName>
    </Property>
    <Property>
      <PropName>/P/H1/</PropName>
      <NvpName>Heading1</NvpName>
    </Property>
    <Property>
      <PropName>/P/H2/</PropName>
      <NvpName>Heading2</NvpName>
    </Property>
    <Property>
      <PropName>/P/H3/</PropName>
      <NvpName>Heading3</NvpName>
    </Property>
    <Property>
      <PropName>/P/Barcode/</PropName>
      <NvpName>Barcode</NvpName>
    </Property>
    <Property>
      <PropName>/P/Table//</PropName>
      <NvpName>Table</NvpName>
    </Property>
    <Property>
      <PropName>/P/Table/Row/Cell</PropName>
      <NvpName>Cell</NvpName>
    </Property>
    <Property>
      <PropName>/P/Para/</PropName>
      <NvpName>Body</NvpName>
    </Property>
  </NvpMapping>
</Filter>
```

- d. Search for the XML element AllWorkflows.

e. Add the following Workflows element to the AllWorkflows node:

```
<Workflows>
  <AllMetaData>
    <TP_Extension>xmlcdc</TP_Extension>
  </AllMetaData>
  <FilterName>XPATH-DATACAP</FilterName>
</Workflows>
```

f. Save and close the file.

Parent topic: [Datacap application development](#)

Application translation

You can translate the text in Datacap applications that is displayed in Datacap clients: Datacap Desktop, FastDoc (Job Monitor only), and Datacap Navigator. The following text can be translated: workflow names, job names, task names, shortcuts, descriptions, field names, document types, page types, and validation error messages.

Application translation is optional. If an application is not translated, or if no default translation is provided and the client's locale or language resource is not found, the application continues to work as expected. Ideally, a translated application should have a resource file in its DCO application root folder to provide a default, but this is not a requirement.

Translation resource files

Translated text is defined in resource files and substituted by Datacap clients. All Datacap clients share the same translation resource files. The resource files, named `resources.json`, are stored in the DCO application folder, for example: `C:\Datacap\MyApplication\dco_MyApplication\resources.json`.

You can create subfolders for translation into multiple languages. Datacap clients look for translation resource files in the following order:

1. `dco_application\locale\resources.json`
2. `dco_application\language\resources.json`
3. `dco_application\resources.json`

locale is in "ll-cc" format, where "ll" is a two-character language code (for example, en) and "cc" is a two-character country code (for example, us).

language is in "ll" format, where "ll" is a two-character language code (for example, en).

Resource file format

The translation resource files contain key-value pairs in the following format:

```
{
  "key1" : "value1",
  "key2" : "value2",
  ...
  "keyN" : "valueN"
}
```

Each key consists of a type and name as follows: `type.name`

type can have one of the following values:

workflow
task
job
jobdescription
shortcut
shortcutdescription
doctype
pagetype
field
message

name refers to the name of the object.

For example, you can define translation text for an ItemDesc field as follows:

```
"field.ItemDesc": "Item description"
```

For more examples, see the sample resource files that are installed with the TravelDocs application in C:\datacap\TravelDocs\dco_TravelDocs.

Important:

- Keys are case-sensitive. The *name* must match the name of the object. For example, `job.Main Job` is not the same as `job.MainJob`.
- Field objects can have a `label` property. If translation text is defined for that field in a resource file, the translation text is displayed instead of the label text. Precedence is given to display text as follows:
 1. Resource value
 2. Label value
 3. Name or ID

The display precedence applies to fields on Verify panels, the Start Batch panel, and the Edit Batch Properties panel.

- If you are using Datacap Navigator, reload the page in the browser to see your changes.

Message translation

The key for a message resource is `message.MsgID`, where *MsgID* is a unique ID that you define to store the message by using the MessageID action. To display a message, the client searches the message object for the MessageID variable. If the variable is not found, the message that is found in the MESSAGE or ErrorMessage variable is displayed without modification. If MessageID is found, the translated message is retrieved from the resource file. Substitution parameters that were provided to the action are replaced in the translated message. The client can also translate substitution parameters by using one of the translated object types such as job, task, or field.

Parent topic: [Datacap application development](#)

Related information:

[MessageID](#)

[MessageIDParameter](#)

Creating a Datacap Maintenance Manager application

Datacap Maintenance Manager applications setup batch monitoring, status notification, and automatic deletion of completed batches.

About this task

Before you create a Datacap Maintenance Manager application, you must be familiar with the following tasks:

- Creating and modifying Datacap applications by using Datacap Studio
- Running Datacap applications by using a Datacap client
- The basic structure of Datacap batches and how batches move through the various stages in the Datacap workflow

You will need to [start or ensure the Datacap Server Service is started](#) before beginning these tasks.

- [Starting Datacap Studio to use the Application wizard](#)
To use the Application wizard to copy an application, start Datacap Studio without opening an application.
- [Creating a Datacap Maintenance Manager application on Datacap Studio](#)
You can use Datacap Studio to create a Maintenance Manager application to set up batch monitoring, status notification, and automatic deletion of completed batches.
- [Updating the datacap.xml file on the Datacap server](#)
The datacap.xml file on the Datacap server must contain references to the applications. The file must also contain the locations of the applications that are indicated in the file and that you are configuring for use.
- [Setting the location of the Datacap Server and the Datacap.xml file](#)
You must define the location of the datacap.xml and the location of the Datacap Server for the application that you want to run.
- [Setting Datacap Maintenance Manager account security permissions for the Datacap\NENU application folder](#)
When the Datacap Server is running Windows, you must set up the appropriate security for the Datacap Maintenance Manager account on the Datacap Server shared c:\Datacap\NENU application folder.
- [Setting Datacap Maintenance Manager account security permissions for the monitored application folder](#)
When the Datacap Server is running Windows, you must set up the appropriate security for the Datacap Maintenance Manager account. Set up this security on the Datacap Server application folder such as c:\Datacap\APT.
- [Opening the Datacap Maintenance Manager application](#)
You use Datacap Studio to open the Datacap Maintenance Manager application.
- [Deleting rulesets from a Datacap Maintenance Manager application](#)
New RRS applications that you create by using the Datacap Studio Application wizard contain a default ruleset structure that monitors only another application. You can delete this default ruleset structure from the Datacap Maintenance Manager application.
- [Adding actions to a Datacap Maintenance Manager application](#)
You can create a simple ruleset with actions that check the status of batches within a specified application workflow and email the results to an administrator.
- [Running a rule set with Datacap Maintenance Manager](#)
You can run a rule set manually on Datacap Maintenance Manager. Running Maintenance Manager manually is useful when you want to test the rule set before you set up Windows Scheduler to run it automatically.
- [Configuring Windows Task Scheduler to automatically run a ruleset](#)
- [AutoDelete batches with Datacap Maintenance Manager](#)
Earlier versions of Datacap included an AutoDelete utility to automatically delete unwanted batches. This function is now available through Datacap Maintenance Manager.

Starting Datacap Studio to use the Application wizard

To use the Application wizard to copy an application, start Datacap Studio without opening an application.

Before you begin

[Start or ensure the Datacap Server Service is started.](#)

Procedure

To start Datacap Studio without opening an application:

1. On the developer workstation, in the Start menu select IBM Datacap Developer Tools > Datacap Studio. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.
2. On the Applications window, click Close. An empty Datacap Studio main window opens on the Rulemanager tab.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

Creating a Datacap Maintenance Manager application on Datacap Studio

You can use Datacap Studio to create a Maintenance Manager application to set up batch monitoring, status notification, and automatic deletion of completed batches.

Procedure

To create a Datacap Maintenance Manager application on Datacap Studio.

1. On the developer workstation, with an empty Datacap Studio main window open, click Datacap application wizard.
2. On the Overview window, click Next.
3. On the Wizard Mode window, select Create a new RRS application and click Next.
4. On the New RRS application window, complete the following fields.

Table 1. Values to enter in the New RRS application window

Field	Entry	Description
Application Name	Maintenance Manager	Name of the application, becomes the name of the folder in which the application is created.
Datacap Folder	C:\Datacap	This entry creates the application shortcut to tmclient.exe.
Destination	\\Server\Datacap	Location where the application files are created.

5. Click Next.
6. At the Document Hierarchy window, click Next.
7. At the Fingerprints window, click Next.
8. At the Add sample images window, click Next.
9. At the Finish window, click Finish to begin the creation process. The Application Wizard - Summary window displays some of the following messages:
 - o General: lists the results that completed successfully
 - o Warnings: something unexpected was encountered, but you can continue
 - o Errors: a problem that must be corrected was encountered
10. Click View Logs to open the appwiz.new.log file and search for any error and warning messages.
11. Close the log file.
12. Click Close to close the Application wizard.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

Updating the datacap.xml file on the Datacap server

The datacap.xml file on the Datacap server must contain references to the applications. The file must also contain the locations of the applications that are indicated in the file and that you are configuring for use.

About this task

Before you begin, identify the names and locations of the applications that you want the Datacap Application Service to be aware of.

Important: The Datacap Application Service is case-sensitive. When you add or change entries in the datacap.xml file, make sure that the case matches the case of the UNC paths, folders, and file names.

Procedure

1. Open the C:\Datacap\datacap.xml file in a text editor on the Datacap server. It contains lines that look like this example:

```
<datacap ver="8.0">
  <app name="Flex" ref="Flex"/>
  <app name="TravelDocs" ref="\\ServerName\Datacap\TravelDocs"/>
  <app name="APT" ref="APT"/>
  <app name="AppWizard" ref="DStudio\AppWizard"/>
  <app name="NENU" ref="NENU"/>
</datacap>
```

2. To delete references to applications that do not exist or are not configured, delete the corresponding <app name= lines. When the only application on the Datacap server is the TravelDocs application, delete all of the lines except the line for the TravelDocs application. The result looks like this example:

```
<datacap ver="8.0">
  <app name="TravelDocs" ref="\\ServerName\Datacap\TravelDocs"/>
</datacap>
```

3. To add references to applications that exist, add a line with the name of the application and the full UNC path to the application folder. When you add a line for the Datacap Maintenance Manager application, your result looks like this example:

```
<datacap ver="8.0">
  <app name="TravelDocs" ref="\\ServerName\Datacap\TravelDocs"/>
  <app name="NENU" ref="\\ServerName\Datacap\NENU"/>
</datacap>
```

4. Save and then close the datacap.xml file.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

Setting the location of the Datacap Server and the Datacap.xml file

You must define the location of the datacap.xml and the location of the Datacap Server for the application that you want to run.

Procedure

To define the location of the server and the datacap.xml file:

1. On the developer workstation, in the Start menu click IBM Datacap Services > Datacap Application Manager. When User Account Control (UAC) is on, the User Account Control window opens. Click Yes.

2. Select the application to which you want to set the location, such as TravelDocs, or APT. The paths are displayed in the fields on the Main tab.
3. Ensure that all workflows are displayed. Check that all of the paths reflect the correct UNC paths to the various files and databases by using the Datacap Server name rather than C:\.
4. Click Locale. Select the option that is associated with the language and regional settings that are used on most of the documents to be processed by the application. If you do not select a locale, the value that is set on the Regional and Language property sheet of the operating system is used.
5. Click the Datacap tab and change the Name or IP address field to the IP address or the name of the Datacap Server without using backslashes.
6. In the Protocol field, select the TCP/IP version that is used on your network. If you do not select a protocol, Datacap defaults to TCP IPv4.
7. Click the Service tab and verify that the path reflects the correct UNC location of the datacap.xml on the server, such as \\Server\Datacap\datacap.xml.
8. Close Datacap Application Manager.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

Setting Datacap Maintenance Manager account security permissions for the Datacap\NENU application folder

When the Datacap Server is running Windows, you must set up the appropriate security for the Datacap Maintenance Manager account on the Datacap Server shared c:\Datacap\NENU application folder.

About this task

The other accounts were already granted security permissions during the installation and configuration of Datacap.

Procedure

To set Maintenance Manager account security permissions for the Datacap\NENU application folder:

1. On the Datacap Server computer, start Windows Explorer.
2. Right-click on the C:\Datacap\NENU folder and select Properties.
3. On the Properties dialog, click the Security tab.
4. Verify that the following accounts are set properly.

Domain/Windows Accounts for	Set to allow
Developers	Full Control
Datacap Server service	Full Control
Maintenance Manager	Read & Execute

5. Close the Properties dialog.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

Setting Datacap Maintenance Manager account security permissions for the monitored application folder

When the Datacap Server is running Windows, you must set up the appropriate security for the Datacap Maintenance Manager account. Set up this security on the Datacap Server application folder such as

c:\Datacap\APT.

About this task

The other accounts were granted security permissions during the installation and configuration of Datacap.

Procedure

To set Maintenance Manager account security permissions for the application folder:

1. On the Datacap Server computer, start Windows Explorer.
2. Go to c:\Datacap*Application*, right-click on the c:\Datacap*Application* folder and select Properties.
3. On the Properties dialog, click the Security tab.
4. When Maintenance Manager is monitoring the application, the Maintenance Manager Domain/Windows account must be set to allow Read & Execute.
5. When Maintenance Manager is deleting batch folders or changing the application, set the Maintenance Manager Domain/Windows account to allow Full Control.
6. Close the Properties dialog.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

Opening the Datacap Maintenance Manager application

You use Datacap Studio to open the Datacap Maintenance Manager application.

Before you begin

[Start or ensure the Datacap Server Service is started.](#)

Procedure

To open the Maintenance Manager application:

1. On the developer workstation, in the Start menu, click IBM Datacap Developer Tools>Datacap Studio. Click Yes at the User Account Control window.
2. Select the Maintenance Manager application, and click Next.
3. At the Datacap Login window, type the default Datacap Admin user ID, Password, and Station ID.
4. Click Finish. Datacap Studio opens with the Rulemanager tab open.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

Deleting rulesets from a Datacap Maintenance Manager application

New RRS applications that you create by using the Datacap Studio Application wizard contain a default ruleset structure that monitors only another application. You can delete this default ruleset structure from the Datacap Maintenance Manager application.

Procedure

To delete the default rulesets from a Maintenance Manager application:

1. With the new application open in Datacap Studio, click the first ruleset on the list, such as VScan.
2. Click Lock/Unlock ruleset. A red x displays next to the selected ruleset.
3. Click Remove object.
4. At the Confirm deletion prompt, click Yes. A warning message indicates that the Document Hierarchy contains references to a nonexistent ruleset.
5. Click OK. The message box closes and the ruleset is deleted.
6. Repeat steps 2 - 5 to delete all of the rulesets.
7. On the Document Hierarchy tab, click Lock DCO for editing. You are prompted to confirm the deletion of the references that are not found.
8. Click Yes. The blue open lock icon changes to orange and locked.
9. Click Save changes.
10. Click Unlock DCO. The orange locked icon changes to blue and unlocked.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

Adding actions to a Datacap Maintenance Manager application

You can create a simple ruleset with actions that check the status of batches within a specified application workflow and email the results to an administrator.

Procedure

To add actions to a Maintenance Manager application:

1. In Datacap Studio, on the Task Profiles tab, click Lock/Unlock task profiles to lock the task profiles list for editing. The blue unlocked icon changes to orange and locked.
2. Click Add a new task profile to create a new task.
3. Select Custom, enter a name for the new task such as Maintenance Manager, and click OK. The window closes and the new task name is displayed in the Task Profiles tab.
4. Click Save changes to save the task profiles list.
5. On the Rulesets tab, right-click the workflow name and choose Add Ruleset. Ruleset1 is added.
6. Click the Ruleset1 label, and rename the new ruleset, such as the name Maintenance Manager.
7. On the Task Profiles tab, select the Maintenance Manager task.
8. On the Rulesets tab, select the Maintenance Manager ruleset and click Add ruleset to profile.
9. On the Task Profiles tab, click Save changes and then click Lock/Unlock task profiles to unlock the task profiles list. The orange locked icon changes to blue and unlocked.
10. Within the new ruleset, select Function 1.
11. Click the Actions Library tab and expand the Maintenance Manager actions to display the available actions.
12. Use Add to function to add each of the following actions to Function 1:
 - o SetUser
 - o SetPassword
 - o SetStation
 - o SetApplication
 - o SetupOpenApplication
 - o QuerySetStatus
 - o ProcessRunSqlQuery
 - o LogSendEmail
13. In the Rulesets tab, select each of the actions that requires a parameter and enter the parameter value in the Properties tab, then press Enter. The parameter value is displayed in the action on the Rulesets tab. For example:

Table 1. Rulesets tab example parameter values

Parameter	Example value
SetUser	admin
SetPassword	admin
SetStation	1
SetApplication	APT
QuerySetStatus	stopped
LogSendEmail	Enter your email address in the addressTo parameter

Attention: The LogSendEmail action takes multiple parameters but in most cases only the addressTo parameter is required. To get information about the other parameters, select LogSendEmail in the Actions Library tab and click Display information about an action or namespace to view the embedded help.

14. On the Rulesets tab, click Save changes. Then, click Lock/Unlock ruleset and select the Publish ruleset. The orange locked icon changes to blue and unlocked.
15. On the Document Hierarchy tab, click Lock DCO for editing to lock the DCO for editing. The blue unlocked icon changes to orange and locked.
16. Expand the top batch level of the DCO and, in the Open node, select global.
17. On the Rulesets tab, select Rule 1 in the Maintenance Manager ruleset and click Add to DCO.
18. On the Document Hierarchy tab, click Save changes and then click Unlock DCO to unlock the DCO. The orange locked icon changes to blue and unlocked.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

Running a rule set with Datacap Maintenance Manager

You can run a rule set manually on Datacap Maintenance Manager. Running Maintenance Manager manually is useful when you want to test the rule set before you set up Windows Scheduler to run it automatically.

Procedure

To run a ruleset with Maintenance Manager:

1. On the developer workstation, in the Start menu select IBM Datacap Developer Tools > Datacap Maintenance Manager. If the User Account Control window displays, click Yes. The Maintenance Manager window opens.
2. On the Maintenance Manager, click Create. Maintenance Manager generates a default settings file.
3. To modify the default settings, click in the empty field next to the lib label. Then, either select a value from the list, or enter a value. Modify the following values:

Table 1. Modifying the default settings file values

Value	Modification
lib	Select the name of the Maintenance Manager application. This application contains the Maintenance Manager task profile.
tprofile	Select the name of the Maintenance Manager task profile.
action_log_level	Select the logging level for action messages. 0 provides maximum information.

Value	Modification
log_override	Select True to create a new log file; False to append to the existing log file.
log_reflush	Select True to ensure that all messages are written to the log even in the case of an exception; runs slower but easier to debug.
service_log	Select the logging level for service messages. 3 provides the maximum batch log information, 0 provides the minimum log information.

4. Select Place settings file in the batch directory. This setting creates a subfolder beneath the Batches folder of the application for the Maintenance Manager working files.
5. Click Save to generate the settings file. The Settings.xml file is saved in the Maintenance Manager folder in the Batches folder of the selected application.
6. Click Run Profile to test the task profile. A message confirms that the task was completed and instructs you to check the log file.
7. Click OK.
8. Using Windows Explorer, open the Maintenance Manager folder under the Batches folder of the application. For example, open C:\Datacap\NENU\batches\NENU_NENU.
9. Review the log file, such as nenu_rrs.log, to see the results of the profile run.
10. If you configured the Maintenance Manager rule set to send an email, check your email inbox for a message that contains the contents of the Maintenance Manager log file.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

Configuring Windows Task Scheduler to automatically run a ruleset

About this task

You can set up a scheduled Windows task to run the Datacap Maintenance Manager task profile that you created. You can also run the security options to set to ensure that the task runs successfully. These instructions pertain to systems that are running on Windows. You must use the Windows Scheduler that is on the computer where the Maintenance Manager component NENU.exe is installed.

Procedure

To configure Windows Task Scheduler to automatically run a ruleset:

1. From the Start menu, select Administrative Tools > Task Scheduler.
2. Under Task Scheduler (Local), select Task Scheduler Library and choose New Folder from the Actions panel.
3. Enter `Datacap` and click OK. The new folder is created for your Datacap scheduled tasks.
4. Expand Task Scheduler Library and select the Datacap folder.
5. In the Actions panel, click Create Basic Task.
6. In the Create a Basic Task dialog, enter a name for the task and click Next.
7. In the Task Trigger dialog, select Daily and click Next.
8. In the Daily dialog, enter the Start date and time, and click Next.
9. In the Action dialog, select Start a program and click Next.
10. In the Program/script field, browse and select C:\Datacap\Taskmaster\NENU.exe, then click Open.

11. In the Add arguments field, enter the path and file name for the Maintenance Manager settings file. For example, enter C:\Datacap\NENU\Batches\NENU_NENU\Settings.xml
12. Click Next.
13. In the Summary dialog, select Open the Properties dialog for this task when I click Finish and then click Finish. The Maintenance Manager Properties dialog opens.
14. If the Properties window for the Maintenance Manager task is not already open, double-click the task name in the Task Scheduler Library. The Properties window opens with the General tab displayed.
15. Under Security Options, identify the domain/Windows account currently associated with the task. If it is not the correct account, click Change User or Group and select the Maintenance Manager domain/Windows account.
16. Select Run whether user is logged on or not.
17. Select Run with highest privileges option.
18. Click OK.
19. Ensure that the Maintenance Manager domain/Windows account is correct, enter the password, then click OK.
20. Click OK to close the Properties window.
21. Close the Task Scheduler.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

AutoDelete batches with Datacap Maintenance Manager

Earlier versions of Datacap included an AutoDelete utility to automatically delete unwanted batches. This function is now available through Datacap Maintenance Manager.

As you process batches, the batches folder of the Datacap application can become large. A batch can lose its value after the Export task copies the verified data to an export file or database in another location. You can create a Maintenance Manager ruleset to periodically purge old batches from the batches folder of the application. The ruleset can also remove the corresponding records from the Engine database of the application.

To create an AutoDelete ruleset, you must complete the following tasks:

- Locate batches that are completed within a specified timeframe, for example, more than five days ago.
- Delete those batches from the batches folder of the application or move them to an archive folder.
- Delete the records that relate to the deleted batches from the Engine database of the application or move those records to a separate archive database.
- Generate a log file that documents the results of the process.

You can then schedule the AutoDelete ruleset to run automatically as needed.

- [AutoDelete Process](#)
To delete old batches from the system, you must remove the information from the batches folder and the Engine database.
- [Sample AutoDelete ruleset](#)
The sample AutoDelete ruleset removes all of the batches that were completed more than five days ago.

Parent topic: [Creating a Datacap Maintenance Manager application](#)

AutoDelete Process

To delete old batches from the system, you must remove the information from the batches folder and the Engine database.

When you run a batch through the application workflow, Datacap does the following tasks.

- Updates the image files, page data, and other items in the batches folder of the application
- Updates the batch status information in the Engine database of the application

Remove the information from the batches folder and the Engine database to delete batches from the system. You can back up the data in another location.

If you want to move the batches or the batch information in the Engine database as part of the deletion process:

- Use the ProcessMoveBatches action to move the batch folders to any available local drive or network share.
- You can move the database information to another database with the same format as the Engine database. The easiest way to create the database is to create another Datacap application use the using the ProcessMoveDBRecords action to move the records to that database.

Parent topic: [AutoDelete batches with Datacap Maintenance Manager](#)

Sample AutoDelete ruleset

The sample AutoDelete ruleset removes all of the batches that were completed more than five days ago.

This ruleset uses a separate Datacap application named BackupApp as an archive repository for the batches and database records that are removed from the production application. APT is the application that is used in this example.

Each of the actions in the sample AutoDelete ruleset is described in the following table.

For more information about Datacap Maintenance Manager actions, see the embedded help in Datacap Studio. To access the embedded help, select an action in the Actions Library tab and click information.

Table 1. Sample AutoDelete ruleset actions

Action	Description
AutoDelete Rule1 Function1	Set application to APT
SetApplication("APT");	
SetUser("admin");	Set user name
SetPassword("admin");	Set password It is best not to store User or Password values in the application without encryption. Use Smart parameters, which are encrypted.
SetStation("1");	Set station
SetupOpenApplication("APT");	Connect to APT application
QuerySetStatus("Job done");	Locate batches with status "Job done"
QuerySetAge("432000", False);	Locate batches completed more than 5 days ago, age is specified in seconds
ProcessRunSQLQuery();	Run query
ProcessMoveBatches("G:\BackupApp\batches");	Move batch folders to back up location

Action	Description
ProcessMoveDBRecords("ICRC_APT", " ", " ", " ", True, "admin", "admin", "1", True);	Move database records to back up application database

Parent topic: [AutoDelete batches with Datacap Maintenance Manager](#)

Datacap object API reference

You can use Datacap object APIs to create or modify runtime batches and document hierarchies, and to obtain or modify recognition confidence levels, field values, text values, and object types. To use Datacap object APIs, you reference TDCO.DLL in your development environment, such as Microsoft Visual Studio.

Datacap contains three separate classes or APIs that you can use to complete actions on different parts of a Datacap object (DCO), which can be a batch, document, page, field, or character. You use the three sets of Datacap API to modify the document hierarchy (setup DCO) and the runtime batch hierarchy (runtime DCO).

Datacap requires an XML file to process a batch. When you use Datacap Studio to create an application, Datacap saves the setup DCO as an XML file, for example, C:\Datacap*application name*\dco_*application name*\dco_*application name*.xml. Similarly, if you create a document hierarchy outside of Datacap Studio, you use an API to save the setup DCO as an XML file. Each Datacap object is represented as a node in the XML file. The setup DCO defines the expected structure of a batch, including valid documents, pages in documents, the fields on each page, and other predefined information.

Datacap creates the runtime DCO when a workflow is started. When a task completes, Datacap writes the batch to a batch folder, for example, C:\Datacap\APT\batches*batch number*. The runtime information comprises a root file that is named after the completed task, for example, C:\Datacap\APT\batches*batch number*\Verify.xml and defines the documents and pages in the batch. Datacap also writes a data file for each page, for example, C:\Datacap\APT\batches*batch number*\tm000001.xml. The runtime DCO indicates the current composition of the batch, including documents, pages, fields, and characters.

The three sets of APIs are DCO, DCOSetup, and DCOSetupNode. You can use the APIs to search or browse through a DCO object, starting from the root node (batch) to a child node, which includes documents, pages, fields, and characters. Using the APIs, you can find and select a child node, and modify or populate a value. You use the DCO APIs with a runtime DCO, the DCOSetup APIs with the setup DCO, and the DCOSetupNode APIs with individual nodes in the setup DCO. The following tasks represent a sample of how you can use some of the APIs:

- Create a document hierarchy
- Modify a batch by creating documents, reorganizing pages, or deleting documents
- Determine the page types that contain confidential information and, with the implementation of custom verification panels, display only those pages to an operator who has appropriate security clearance
- Create, modify, or populate fields and characters
- Access and modify a dictionary that can be used to present a list of valid field values to an operator during a verification task
- Use XML files to load a batch, or save a batch to a disk
- [Relationship between runtime batch and document hierarchy](#)
To know which APIs to use for specific objects, you need to understand how a runtime batch and a document hierarchy are related. For example, when the page types are identified in a runtime batch, you can insert document objects into the batch by referencing the DCO setup object.
- [Creating a document hierarchy with Datacap object APIs](#)
You can use the Datacap object APIs to create a document hierarchy. For example, you can use these APIs to read a setup DCO XML file and add child objects, including documents, pages, and fields.

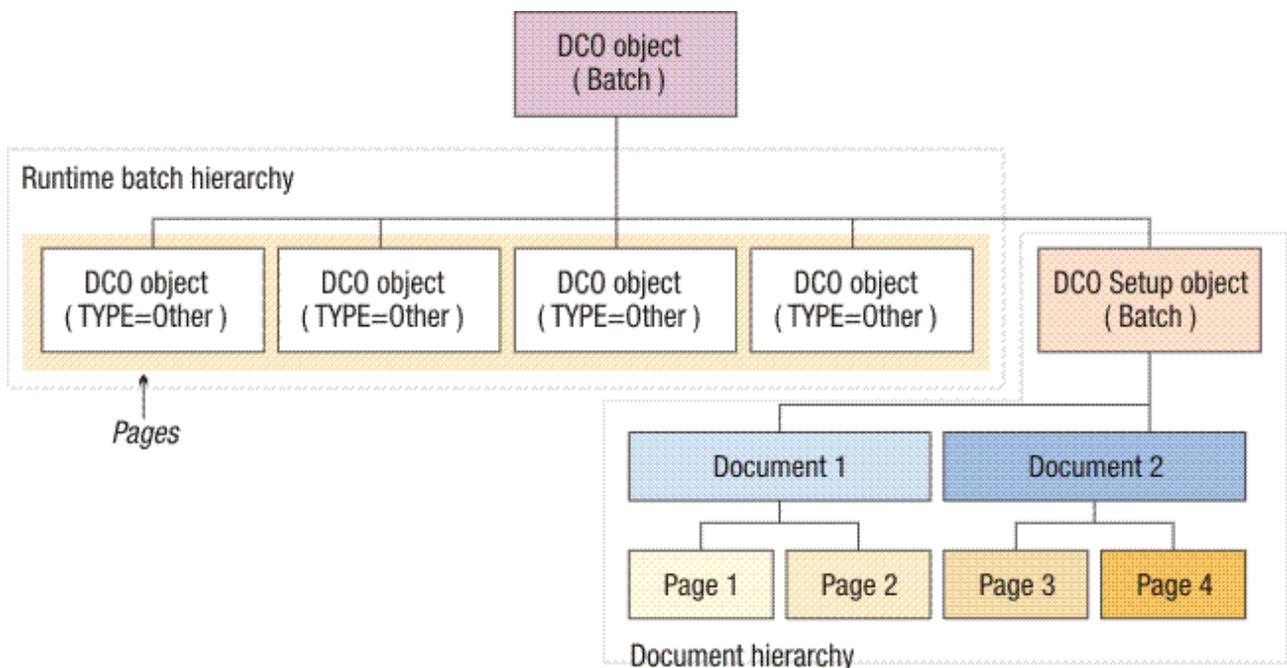
- [DCO APIs](#)
The DCO APIs include properties and methods that you can use to modify the runtime batch hierarchy, which is also known as the runtime DCO. For example, when your business requirements change, you can use these APIs to add or remove documents and pages, or change the names or values of object types.
- [DCOSetup API](#)
The DCOSetup APIs include properties and methods that you can use to create or modify the document hierarchy. Datacap saves the document hierarchy as the setup DCO file in XML format, for example, C:\Datacap\application name\dco_application name\application name.xml.
- [DCOSetupNode APIs](#)
You can use the DCOSetupNode API properties and methods to access and modify Setup DCO child objects, including rules and variables.

Related concepts:
[Document hierarchy](#)

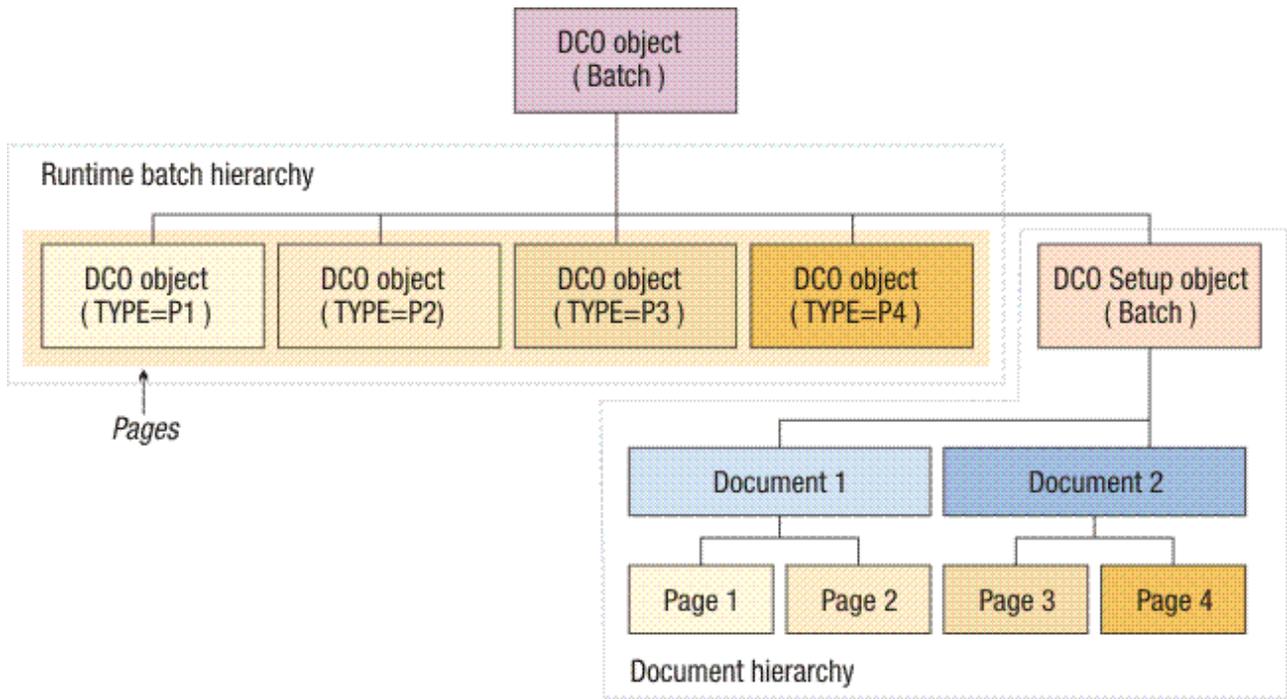
Relationship between runtime batch and document hierarchy

To know which APIs to use for specific objects, you need to understand how a runtime batch and a document hierarchy are related. For example, when the page types are identified in a runtime batch, you can insert document objects into the batch by referencing the DCO setup object.

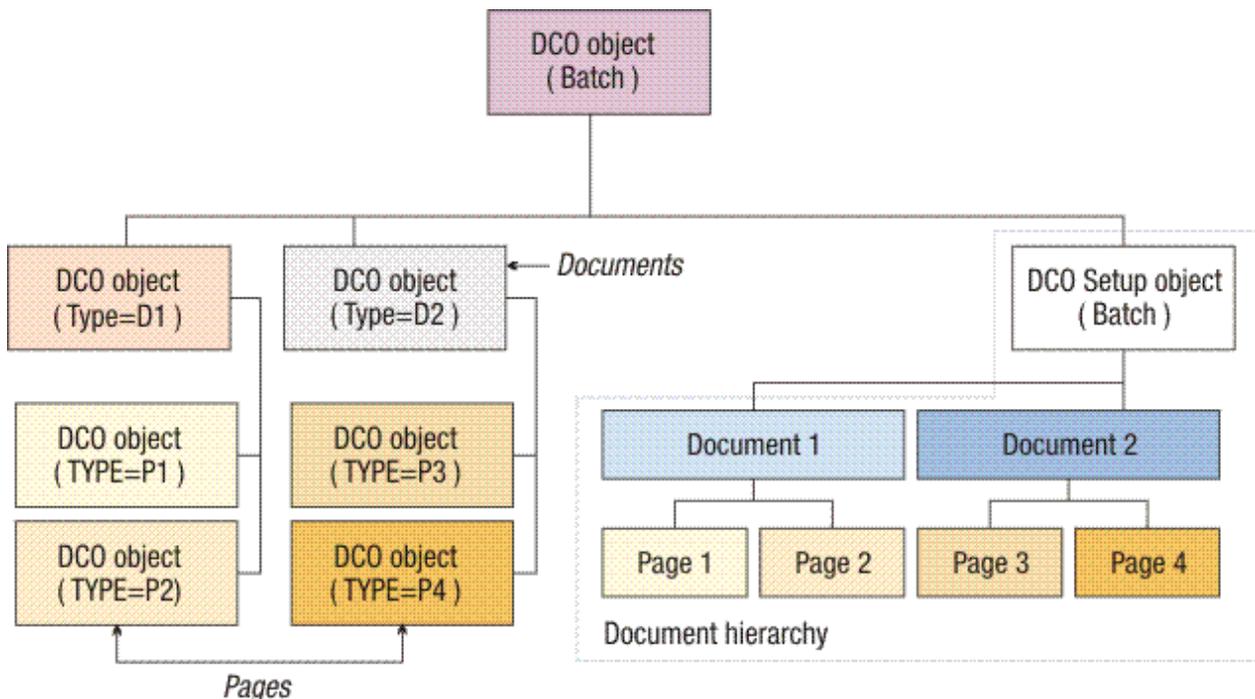
At the beginning of a workflow, Datacap scans a collection of pages. Because the page types are initially unknown, Datacap assigns the type `Other` to the pages when it adds them to the runtime batch hierarchy. The following diagram shows that the document hierarchy includes four pages. At run time, the four pages are initially assigned the type `Other` when Datacap adds the pages to a batch.



Datacap identifies each page type by referencing the types that are defined in the document hierarchy, and assigns the appropriate `TYPE` attribute to each page object. For example, the four pages in the runtime batch hierarchy are identified as `P1`, `P2`, `P3`, and `P4`, as shown in the following diagram:



Datacap gets the document type that is associated with each page type and inserts the appropriate document objects into the runtime hierarchy. Previously, the batch included the four pages without the associated document type. The following diagram shows that P1 and P2 are associated with D1, and P3 and P4 are associated with D2. The page and document type association is based on the document hierarchy that is defined in the DCO Setup object:



Parent topic: [Datacap object API reference](#)

Creating a document hierarchy with Datacap object APIs

You can use the Datacap object APIs to create a document hierarchy. For example, you can use these APIs to read a setup DCO XML file and add child objects, including documents, pages, and fields.

Before you begin

This task also assumes that you successfully exported a batch of documents from a database to an XML file.

About this task

Because the use of Datacap object APIs is limited to runtime batches and document hierarchies, the task does not include steps for data recognition and initial calculation of confidence levels.

Procedure

To create a document hierarchy by using Datacap object APIs, complete the following steps:

1. Use the ReadSetup method read a document hierarchy that you exported:

```
m_oDCO.ReadSetup("C:\\temp_location\\exported_application.XML");
```

2. Use the WriteSetup method to populate your new setup object and save the document hierarchy as an XML file:

```
m_oDCO.WriteSetup("C:\\Datacap\\NEW\\dco_NEW\\NEW.XML")
```

3. Optional: Use the AddNode method to add child objects (documents, pages, or fields) to the setup object.

TheAddNode method uses two arguments: nType lpszNodeName. The nType value (1) specifies that the new node is a document object. The lpszNodeName value specifies that the document object name is NewNode, as shown in the following example:

```
m_oDCOSetup.AddNode(1, "NewNode");
```

The resulting node in the DCO Setup XML file is <D type="NewNode">.

Tip: You can similarly add pages, fields, or characters by setting the nType value to 2 (for pages), 3 (for fields), or 4 (for characters).

4. Use the AddRule method to specify the structure of a document. This step gets the SetupNode object for the NewNode document in the Setup DCO. The step then adds a rule that requires one instance of the page NewPage to exist for the document structure to be valid. If a page node for NewPage does not exist, the method creates NewPage automatically.

```
TDCOLib.DCOSetup m_oDCOSetup = m_oDCO.SetupObject();
TDCOLib.DCOSetupNode m_oDCOSetupNode =
    m_oDCOSetup.GetNodeByName(1, "NewNode");
m_oDCOSetupNode.AddRule(2, "NewPage", 0, 1, 1);
```

After the execution of the code, the following line is added to the Setup DCO beneath the NewNode document type.

```
<D type="NewNode">
.
.
.
<P type="NewPage" pos="0" min="1" max="1"/> <!-- New line added -->
```

Additionally, if NewPage does not exist, the following page node is created in the Setup DCO:

```
<P type="NewPage">
    <V n="ID">0</V>
    <V n="TYPE">Page</V>
    <V n="STATUS">0</V>
```

```

    <V n="IMAGEFILE"></V>
    <V n="DATAFILE"></V>
    <V n="TEMPLATE IMAGE"></V>
    <V n="MIN_TYPES">0</V>
    <V n="MAX_TYPES">0</V>
</P>

```

5. Use the Read method, Write method, and the XML property to export the document hierarchy to an external location, which can be a content repository, a database, or a web server. This example exports the document hierarchy to a temporary location on a local server:

```

m_oDCO.Read("C:\\C:\\Datacap\\NEW\\dco_NEW\\NEW.XML");
strDCOXml = m_oDCO.XML;
m_oDCO.Write ("c:\\temp_location\\new_application.xml")

```

The Read method stores the document hierarchy in a setup object. The XML property assigns the XML file to a variable, and the Write method writes the document hierarchy (setup DCO) to a new location.

Parent topic: [Datacap object API reference](#)

Related reference:

[WriteSetup method](#)

[AddNode method](#)

[AddRule method](#)

DCO APIs

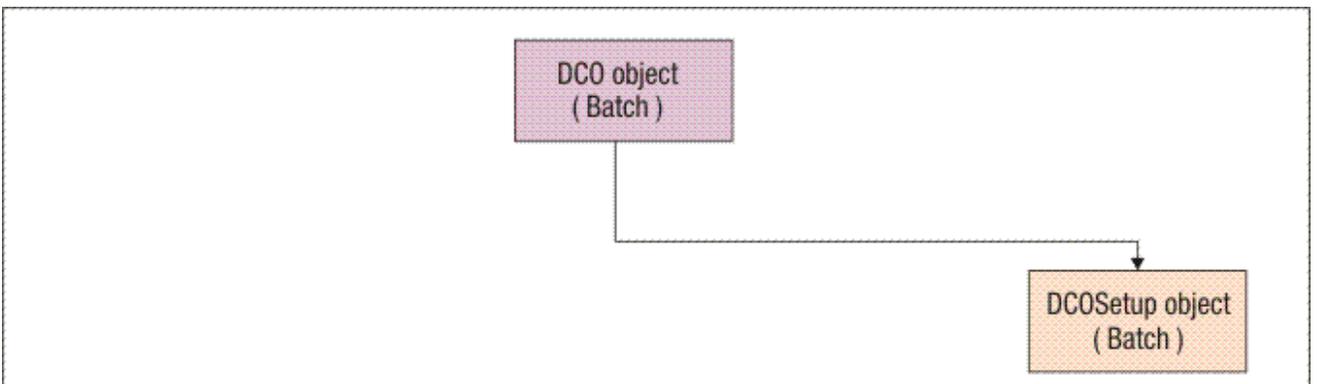
The DCO APIs include properties and methods that you can use to modify the runtime batch hierarchy, which is also known as the runtime DCO. For example, when your business requirements change, you can use these APIs to add or remove documents and pages, or change the names or values of object types.

When you create a runtime DCO object (`m_oDCO` in this example), a corresponding `DCOSetup` object is created automatically. The `DCOSetup` object is a child of the DCO object.

```

TDCOLib.IDCO m_oDCO = new TDCOLib.DCOClass();
//Creating a runtime DCO object

```

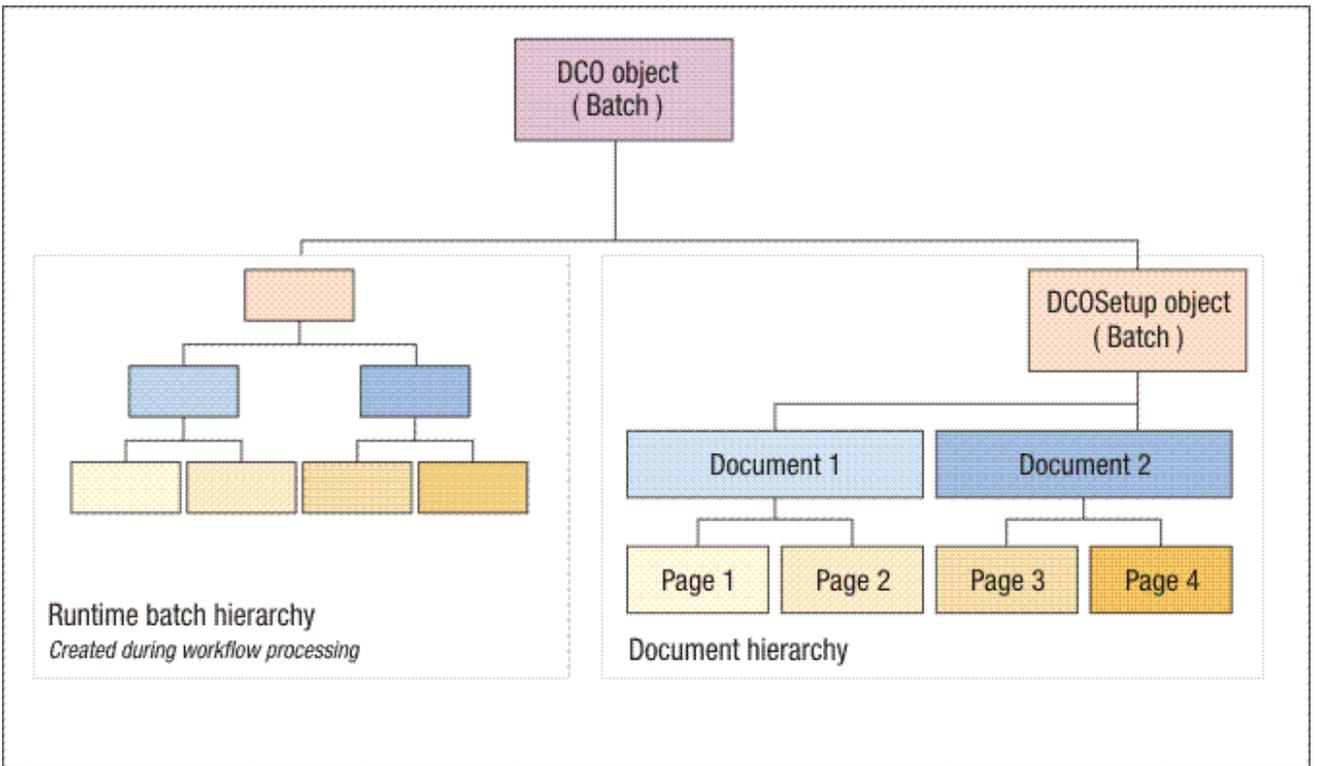


Because the `DCOSetup` object is empty, you need to populate the `DCOSetup` object from the Setup DCO file of the application.

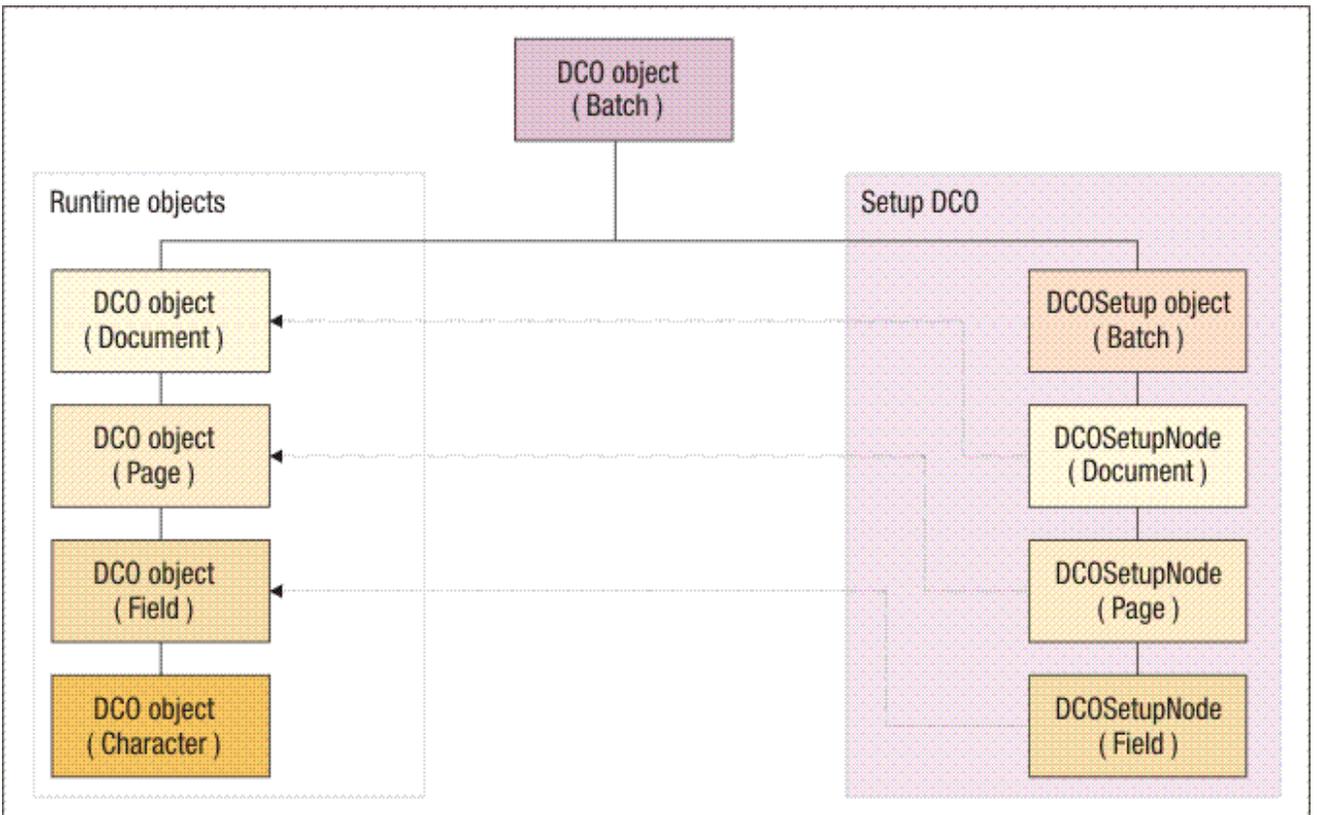
```

m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
//Reading a Setup DCO file

```



As workflow tasks process scanned pages, Datacap creates a hierarchy of child objects beneath the runtime DCO object. Each child object in the runtime batch hierarchy corresponds to an object in the Setup DCO:



You use the DCO API properties to change the names of object types, and you can use methods to add or remove child objects.

- [DCO properties](#)

You can use DCO properties to access or modify the names and values of DCO objects, including

documents, pages, fields, and characters. You can also use DCO properties to modify metadata, which includes confidence levels, IDs, types, statuses, and variables. Generally, you use the properties to pass values to methods that complete actions on the associated objects.

- [DCO methods](#)
You can use DCO methods to complete actions on DCO objects, such as adding documents to batches, pages to documents, or fields to pages. You can also use the DCO methods to search for and change metadata values, including confidence levels, indexes, variables, and alternative text for verification.

Parent topic: [Datacap object API reference](#)

DCO properties

You can use DCO properties to access or modify the names and values of DCO objects, including documents, pages, fields, and characters. You can also use DCO properties to modify metadata, which includes confidence levels, IDs, types, statuses, and variables. Generally, you use the properties to pass values to methods that complete actions on the associated objects.

- [AltConfidenceString property](#)
The AltConfidenceString property sets or gets the confidence level of characters in the field value.
- [AltText property](#)
The AltText property sets or gets the alternative character data that is associated with a field. You use this property with multiple-pass verification tasks and when you need to display alternative text options to an operator.
- [CharConfidence property](#)
The CharConfidence property sets or gets the confidence level of the value of a character.
- [CharValue property](#)
The CharValue property sets or gets the data value of a character, which can be a digit in a multiple-digit number or a single character in a string. You can use this property to search for or assign a value to a character in a field.
- [ConfidenceString property](#)
The ConfidenceString property sets or gets the confidence values for the primary character data in a field object.
- [ID property](#)
The ID property sets or gets the unique identifier of an object, which can be a batch, document, or page.
- [ImageName property](#)
The ImageName property sets or gets the full path and the file name of the image file that is associated with a page object. You can use this property when you are modifying the order of pages in a batch.
- [Status property](#)
The Status property sets or gets the status of an object. The status is a numerical value that indicates success, failure, error, problem, or other status that depends on the type of object. For example, you can use this property to locate pages with a status of 1, which indicates that there was a problem during the page identification task.
- [Text property](#)
The Text property sets or gets the primary character data that is associated with a field.
- [Type property](#)
The Type property sets or gets the object name. You can use this property to search for document types or page types in a batch and change the name of the object.
- [Variable property](#)
The Variable property sets or gets the value of a named variable. You can use this property when you first need to obtain a value that you pass afterward to another variable or object, such as a field.
- [XML property](#)
The XML property sets or gets the XML file that is associated with an object. You can use this property to populate a run time object from the Setup DCO file and assign the XML file to a variable.

Parent topic: [DCO APIs](#)

AltConfidenceString property

The AltConfidenceString property sets or gets the confidence level of characters in the field value.

VBScript restriction: Because extended properties are not supported through the C# .NET Interop interface, you must instead use the [set_AltConfidenceString](#) method or the [get_AltConfidenceString method](#) method.

Syntax

VBScript

```
oDCO.AltConfidenceString (nIndex as Long) as String.
```

Applies to

Field objects only.

Type

Read and write.

Arguments

nIndex

0 specifies the primary value. Where the field contains multiple recognition, voting, or multi-pass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

Return value

String that contains the confidence level of each character.

VBScript example

The following example sets the confidence string for the first alternative in the Total field:

```
objTotalField.AltConfidenceString(1) = "987999"
```

In the example, the Total field consists of 6 characters, and the confidence level of each character is 9, 8, 7, 9, 9, and 9.

Parent topic: [DCO properties](#)

Related reference:

[AltText property](#)

[CharConfidence property](#)

[ConfidenceString property](#)

[get_AltConfidenceString method](#)

AltText property

The AltText property sets or gets the alternative character data that is associated with a field. You use this property with multiple-pass verification tasks and when you need to display alternative text options to an

operator.

VBScript restriction: Because extended properties are not supported through the C# .NET Interop interface, you must instead use the [set_AltText method](#) or the [get_AltText method](#).

Syntax

VBScript

```
oDCO.AltText (nIndex as Long) as String
```

Applies to

Field objects only.

Type

Read and write.

Arguments

nIndex

0 specifies the primary value. When the field contains multiple recognition, voting, or multiple-pass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

Returns

String containing the alternative text. The string is stored in the page XML file in ASCII format.

VBScript example

This example shows the first alternative value (231.77) for the Total field that an operator entered during a multiple-pass verification task.

```
objTotalField.AltText(1) = "231.77"
```

Parent topic: [DCO properties](#)

Related reference:

[AltConfidenceString property](#)

[Text property](#)

[CharValue property](#)

CharConfidence property

The CharConfidence property sets or gets the confidence level of the value of a character.

VBScript restriction: Because extended properties are not supported through the C# .NET Interop interface, you must instead use the [set_CharConfidence method](#) or the [get_CharConfidence method](#).

The confidence level of each character is a digit from 0 (lowest confidence) to 9 (highest confidence). If the confidence level for a four-character field is 9999, each of the 4 characters has a confidence level of 9.

Syntax

VBScript

```
oDCO.CharConfidence (nIndex as Long) as Long
```

Applies to

Character objects only.

Type

Read and write.

Arguments

nIndex

0 specifies the primary value. When the field contains multiple recognition, voting, or multiple-pass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

VBScript example

This example sets the confidence level of the primary character to 9, or the highest confidence level.

```
oDCO.CharConfidence(0)=9 ` high confidence  
conf = oDCO.CharConfidence(0)
```

Parent topic: [DCO properties](#)

Related reference:

[CharValue property](#)

[AltConfidenceString property](#)

CharValue property

The CharValue property sets or gets the data value of a character, which can be a digit in a multiple-digit number or a single character in a string. You can use this property to search for or assign a value to a character in a field.

VBScript restriction: Because extended properties are not supported through the C# .NET Interop interface, you must instead use the [set_CharValue method](#) or the [get_CharValue method](#).

Syntax

VBScript

```
oDCO.CharValue (nIndex as Long) as String.
```

Applies to

Character objects only.

Type

Read and write.

Arguments

nIndex

0 specifies the primary value. When the field contains multiple recognition, voting, or multiple-pass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

Parent topic: [DCO properties](#)

Related reference:

[AddValue method](#)

[CharConfidence property](#)

[DeleteValue method](#)

ConfidenceString property

The ConfidenceString property sets or gets the confidence values for the primary character data in a field object.

The value of each character can be 0 - 9 and corresponds to integer confidence levels of 0 to 9. After Datacap scans a page and applies recognition rules, Datacap generates an XML file that contains confidence levels for characters in each field on a page. The XML code adds a value of 1 to the confidence level so that the range in XML is 1 - 10.

In this XML sample for the Vendor field, the fourth character contains a confidence level of 6 for the primary character value. The first alternative character value has confidence level of 8, and the second alternative character value has a confidence level of 4.

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev501_Vendor">
  <V n="TYPE">Vendor</V>
  <V n="Position">0,0,0,0</V>
  <V n="Status">1</V>
  <C cn="6,8,10" cr="0,0,0,0">83,49,53</C>
  <C cn="6,8,10" cr="0,0,0,0">116,50,54</C>
  <C cn="6,8,10" cr="0,0,0,0">105,51,55</C>
  <C cn="6,8,4" cr="0,0,0,0">110,52,56</C>
```

Syntax

VBScript

```
oDCO.ConfidenceString as String
```

C#

```
string ConfidenceString { set; get; }
```

Applies to

Field objects only.

Type

Read and write.

C# example

This example sets and gets the confidence string that is associated with the primary character data in the Vendor field. The root of the search in this case is the page object.

```
m_oDCOPage.FindChild("Vendor").ConfidenceString = "7777";
string strReturn = m_oDCOPage.FindChild("Vendor").ConfidenceString;
```

- The confidence levels that you specify are the internal confidence levels (0 - 9). When you write them to the page XML file, field object and parent page object confidence values are raised by 1 because the XML values range 1 - 10.
- If the field contains more characters than specified when you are setting the confidence string, the last specified character is used for all remaining characters. For example, if you specify 1234 but the field has 10 characters, the remaining 6 characters are assigned a confidence value of 4.

Parent topic: [DCO properties](#)

Related reference:

[AltConfidenceString property](#)

[AltText property](#)

[CharConfidence property](#)

[get_AltConfidenceString method](#)

[get_CharConfidence method](#)

[set_AltText method](#)

[set_CharConfidence method](#)

[Text property](#)

[Write method](#)

ID property

The ID property sets or gets the unique identifier of an object, which can be a batch, document, or page.

This property provides access to the `id` attribute of an object. For example, the `id` attribute for a batch, document, page, and field can be <B

```
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev544_20100096">, <D
```

```
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev544_20100096.001.01">, <P
```

```
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev544_TM000001">, and <F
```

```
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev544_Vendor">. You can pass the value of the id attribute to a variable or custom action.
```

Syntax

VBScript

```
oDCO.ID as String
```

C#

```
string ID { set; get; }
```

Applies to

All objects. The ID property typically has a value for a batch and for all child nodes, including documents, pages, and fields, but not for characters.

Type

Read and write.

Parent topic: [DCO properties](#)

ImageName property

The ImageName property sets or gets the full path and the file name of the image file that is associated with a page object. You can use this property when you are modifying the order of pages in a batch.

This property provides access to the variable *IMAGEFILE*.

Syntax

VBScript

```
oDCO.ImageName as String
```

C#

```
string ImageName { set; get; }
```

Applies to

Page objects only.

Type

Read and write.

C# example

The following example sets and gets the full path and the file name for the image file that is associated with the page object with ID "TM000001":

```
m_oDCO.FindChild("TM000001").ImageName = "C:\\Datacap\\APT\\batches  
\\20100096.001\\tm000001.tif";  
strImageName = m_oDCO.FindChild("TM000001").ImageName;
```

Parent topic: [DCO properties](#)

Status property

The Status property sets or gets the status of an object. The status is a numerical value that indicates success, failure, error, problem, or other status that depends on the type of object. For example, you can use this property to locate pages with a status of 1, which indicates that there was a problem during the page identification task.

This property provides access to the *STATUS* variable, which is one of the standard variables that are installed with Datacap.

Syntax

VBScript

```
oDCO.Status as Long
```

C#

```
int Status { set; get; }
```

Applies to

All object types.

Type

Read and write.

Parent topic: [DCO properties](#)

Related information:

[Standard Variable Reference](#)

Text property

The Text property sets or gets the primary character data that is associated with a field.

The characters of a field are stored as ASCII values in the page data XML file. In this sample XML node for the Vendor field, the fourth character contains a primary text value of 97. The first alternative text value is 52, and the second alternative text value is 97.

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev594_Vendor">
  <V n="TYPE">Vendor</V>
  <V n="Position">0,0,0,0</V>
  <V n="STATUS">1</V>
  <C cn="10,8,10" cr="0,0,0,0">68,49,68</C>
  <C cn="10,8,10" cr="0,0,0,0">97,50,97</C>
  <C cn="10,8,10" cr="0,0,0,0">116,51,116</C>
  <C cn="10,8,10" cr="0,0,0,0">97,52,97</C>
</F>
```

Syntax

VBScript

```
Text as String
```

C#

```
string Text { set; get; }
```

Applies to

Field objects only.

Type

Read and write.

C# example

This example sets the primary character data in the Vendor field to IBM. The confidence values (cn attribute) are automatically set to high (9 in the code; 10 in the page XML file).

```
m_oDCOPage.FindChild("Vendor").Text = "IBM";
```

- The text is the value that a task, such as Profiler, adds to a runtime field object.
- If you use this property to replace existing data, all characters are assigned high confidence, unless you specify otherwise by using the `ConfidenceString` property.

Parent topic: [DCO properties](#)

Related reference:

[AltText property](#)

[ConfidenceString property](#)

Type property

The `Type` property sets or gets the object name. You can use this property to search for document types or page types in a batch and change the name of the object.

This property provides access to the `TYPE` variable, which is one of the standard variables that are installed with Datacap.

Syntax

VBScript

```
objRT.Type as String
```

C#

```
string Type { set; get; }
```

Applies to

Any object, but it is used primarily for documents and pages.

Type

Read and write.

Example

This example sets the name of the document object to 1040ez:

```
oDCO.Type = "1040ez"
```

Parent topic: [DCO properties](#)

Related reference:

[get_Variable method](#)

[set_Variable method](#)

Related information:

[Standard Variable Reference](#)

Variable property

The `Variable` property sets or gets the value of a named variable. You can use this property when you first need to obtain a value that you pass afterward to another variable or object, such as a field.

VBScript restriction: Because extended properties are not supported through the C# .NET Interop interface, you must instead use [set_Variable](#) or [get_Variable](#).

Syntax

VBScript

```
oDCO.Variable(lpszName as String) As Variant.
```

Arguments

`lpszName`

The name of the variable of which you need to set or get the value.

Applies to

All objects.

Type

Read and write.

VBScript example

This example checks the value of the *Handwriting* variable. If *Handwriting* is greater than 0, then the code sets the value of the *Checked* variable to 1 to indicate that a check box is selected.

```
If oDCO.Variable("HandWriting") > 0 Then
    WriteLog("HandWriting is TRUE")
End If
oDCO.Variable("Checked") = 1
```

Parent topic: [DCO properties](#)

Related reference:

[AddVariable method](#)

[GetChild method](#)

[GetVariableValue method](#)

XML property

The XML property sets or gets the XML file that is associated with an object. You can use this property to populate a run time object from the Setup DCO file and assign the XML file to a variable.

Syntax

C#

```
string XML { set; get; }
```

Arguments

None.

Returns

String containing the XML file name.

Applies to

All object types.

C# example

This example populates the runtime batch object from the DCO Setup file, and then assigns the XML file to the variable *strDCOXml*.

```
m_oDCO.Read("C:\\Datacap\\APT\\batches\\20100096.001\\Verify.xml");  
strDCOXml = m_oDCO.XML;
```

Parent topic: [DCO properties](#)

DCO methods

You can use DCO methods to complete actions on DCO objects, such as adding documents to batches, pages to documents, or fields to pages. You can also use the DCO methods to search for and change metadata values, including confidence levels, indexes, variables, and alternative text for verification.

- [AddChild method](#)
The `AddChild` method adds a child object to the parent runtime DCO object. You can use the `AddChild` method to add a field to a page or add a page to a document. You can add only a child that is at the same or a lower level than this object in the hierarchy.
- [AddValue method](#)
The `AddValue` method adds a character data value (ASCII code) and the associated confidence level to the character object. You can use this method to correct optical character recognition (OCR) errors.
- [AddVariable method](#)
The `AddVariable` method adds a variable and its value to an object, such as a batch, document, page, or field. You can use this method when you need to pass values to a custom action. For example, your organization might introduce a new security regulation that requires a new field for identification on certain page types.
- [AddVariableFloat method](#)
The `AddVariableFloat` method adds a variable of type `double` and its value to an object.
- [AddVariableInt method](#)
The `AddVariableInt` method adds an integer variable and its value to an object. You can use this method when you need to pass values to custom actions.
- [AddVariableString method](#)
The `AddVariableString` method adds a string variable and its value to an object. You can use this method when you need to pass values to custom actions.
- [CheckIntegrity method](#)
The `CheckIntegrity` method determines whether a batch conforms to the document integrity rules that are specified in the Setup DCO file.
- [Clear method](#)
The `Clear` method removes all of the child objects of a runtime object, but does not remove the object itself. For example, you can use this method when a document contains corrupted or incorrect pages.
- [CreateDocuments method](#)
The `CreateDocuments` method uses the Setup DCO file to determine the document type that is associated with each page in a batch and creates the required document objects.
- [CreateFields method](#)
The `CreateFields` method creates a runtime field object for each field that is specified within the

corresponding page (or field) node in the document hierarchy.

- [DeleteChild method](#)
The `DeleteChild` method removes the specified child object from the runtime DCO object.
- [DeleteValue method](#)
The `DeleteValue` method deletes a data value and the corresponding confidence level value of a character object.
- [DeleteVariable method](#)
The `DeleteVariable` method removes a variable, including variables such as `TYPE` or `STATUS` that are installed with Datacap, and its value from a runtime DCO object.
- [FindChild method](#)
The `FindChild` method gets an interface to a child object that is referenced by ID (name). You can use this method when you need to get the interface to a child object for which you only know the name.
- [FindChildIndex method](#)
The `FindChildIndex` method returns the index of the specified child object that is contained within a runtime DCO object. You can use this method when you know the name of the child object, and you need to pass the position of the child object to a variable.
- [FindRouteChild method](#)
The `FindRouteChild` method returns the interface to an object that is specified by using the path through the document hierarchy. You use this method to save or restore execution points during an action process.
- [FindVariable method](#)
The `FindVariable` method returns the index of a variable by using the ID (name) of the variable. You can use this method when you need to pass the value of a variable to another variable or action.
- [get_AltConfidenceString method](#)
The `get_AltConfidenceString` method gets the confidence level of each character in the referenced field. You can use this method when you need to pass the confidence level of a string to a variable or action.
- [get_AltText method](#)
The `get_AltText` method gets the primary or alternative character data that is associated with a field.
- [get_CharConfidence method](#)
The `get_CharConfidence` method gets the primary or alternative character data that is associated with a field. You can use this method when you need to pass the character data value to a variable or action.
- [get_CharValue method](#)
The `get_CharValue` method gets the ASCII data value of the primary character or alternative character. You can use this method when you need to pass the character data value to a variable or action.
- [get_OMRValue method](#)
The `get_OMRValue` method gets the positions within the optical mark recognition (OMR) field of the check boxes that are selected. You can use this method to pass the positions to variables or actions, such as those that are used in verification functions to confirm that check boxes are selected.
- [get_Variable method](#)
The `get_Variable` method gets the value of a referenced variable by using the ID (name) of the variable. You can use this method to pass the value of a named variable to an action for further processing.
- [GetChild method](#)
The `GetChild` method gets a referenced child object by using the index of the child object. You can use this method to retrieve a field on a page when you know the position, but not the name.
- [GetLastError method](#)
The `GetLastError` method retrieves the text of the last error that is encountered during a DCO read operation or write operation, and clears the error.
- [GetPosition method](#)
The `GetPosition` method gets the position of a field or character on a page. You can use this method for fields or characters that are located at identical positions on multiple pages in a batch.
- [GetRoute method](#)
The `GetRoute` method gets the path through the document hierarchy to an object.

- [GetVariableName method](#)
The `GetVariableName` method gets the name (ID) of a variable from the index of the variable. You can use this method when you know the position of a variable within an object, but you do not know the name of the variable.
- [GetVariableValue method](#)
The `GetVariableValue` method gets the value of a referenced variable by using the index of the variable.
- [IsError method](#)
The `IsError` method indicates whether an error occurred during a prior `Read` operation or `Write` operation.
- [IsRoute method](#)
The `IsRoute` method indicates whether the specified path is the valid path through the runtime hierarchy for an object.
- [IsValid method](#)
The `IsValid` method confirms that the interface of a DCO object is valid and is connected to an actual object.
- [MoveChild method](#)
The `MoveChild` method moves a child of a runtime DCO object to a different index location. You can use this method to reorganize or correct batches, documents, pages, or fields.
- [MoveIn method](#)
The `MoveIn` method moves the specified runtime DCO object from the current parent to a different parent.
- [NumOfChildren method](#)
The `NumOfChildren` method returns the number of child objects that is associated with the runtime DCO object.
- [NumOfVars method](#)
The `NumOfVars` method returns the number of variables that is associated with the runtime DCO object.
- [ObjectType method](#)
The `ObjectType` method returns a numeric value that indicates the object type.
- [Parent](#)
The `Parent` method returns the parent of the runtime DCO object.
- [Read method](#)
The `Read` method reads the runtime information from a Runtime DCO file or a page data file, and writes the information into the DCO object.
- [ReadSetup method](#)
The `ReadSetup` method reads the document hierarchy setup information from the specified Setup DCO file and writes the information into the Setup object and SetupNode object.
- [SetPosition method](#)
The `SetPosition` method sets the position of the runtime field or character object on the page. This method is useful for making corrections to zonal recognition on a page.
- [SetupNode method](#)
The `SetupNode` method accesses the SetupNode object that is associated with the current object. You can use this method to get the name of the object in the document hierarchy that corresponds to the object in the runtime batch hierarchy.
- [SetupObject method](#)
The `SetupObject` method returns the Setup object that is associated with the current batch-level runtime DCO object. You can use this method when a batch contains unidentified pages and you need to modify the runtime batch hierarchy.
- [set_AltConfidenceString](#)
The `set_AltConfidenceString` method sets the confidence level of each character in the referenced field.
- [set_AltText method](#)
The `set_AltText` method sets the primary character data or alternative character data that is associated with a field. You can use this method for multi-pass verification tasks and double-blind data entry tasks.

- [set_CharConfidence method](#)
The set_CharConfidence method sets the confidence level for the primary value or of the alternative value of a character in a field.
- [set_CharValue method](#)
The set_CharValue method sets the ASCII data value of the primary character or alternative character.
- [set_OMRValue](#)
The set_OMRValue method sets the character values within the OMR field to indicate whether a check box is selected.
- [set_Variable method](#)
The set_Variable method sets the value of a referenced variable by using the name of the variable.
- [Write method](#)
The Write method saves the runtime batch object to a batch file, or a page object to a page file. When it writes a batch object, this method also writes all of the child objects, including documents, pages, and fields.
- [WriteSetup method](#)
The WriteSetup method writes the Setup object and SetupNode objects to the Setup DCO file. You use this method after you read a setup DCO file from an external location.

Parent topic: [DCO APIs](#)

AddChild method

The AddChild method adds a child object to the parent runtime DCO object. You can use the AddChild method to add a field to a page or add a page to a document. You can add only a child that is at the same or a lower level than this object in the hierarchy.

Syntax

VBScript

```
oDCO.AddChild (ObjectType as Long, ID as String,  
Placement as Long) as Object
```

C#

```
TDCOLib.DCO AddChild(int nType, string lpszID, int nIndex)
```

Applies to

All objects.

Arguments

nType

The type of the child object:

- 0=Batch
- 1=Document
- 2=Page
- 3=Field
- 4=Character

lpszID

A string with the ID of the new object. Use an empty string for character objects.

nIndex

The index of the child object relative to other child objects of the same parent. If necessary, existing child objects are moved down to accommodate a new child object.

-1: Adds the new child object to the end of the list.

0: Adds the new child object to the beginning of the list.

Returns

The child object of the specified type.

C# examples

The first example creates a field object under the parent page object. The field is added to the page data file as the last field in the list by using a value of -1.

```
m_oDCOField = m_oDCOPage.AddChild(3, "NewField", -1);
```

The second example adds a character object to the parent field object. The new character is added at the beginning of the list (0) and all existing character fields move down one position.

```
m_oDCOChar = m_oDCOField.AddChild(4, "", 0);
```

- You cannot add a parent to a child. For example, you cannot add a page object to a field object.
- You cannot assign an ID that is already used by another child of the same parent.

Parent topic: [DCO methods](#)

Related reference:

[DeleteChild method](#)

AddValue method

The AddValue method adds a character data value (ASCII code) and the associated confidence level to the character object. You can use this method to correct optical character recognition (OCR) errors.

In addition to the primary data value, each character in a field can have multiple alternative data values. Each of the alternative data values has an associated confidence level (0 - 9 internally and 1 - 10 in the page XML file). Each time that you add a value by using the AddValue method, the new value is appended to the list of existing values. Also, a new confidence level is appended to the list of existing confidence levels.

In the following example, the fourth character contains a primary confidence level value of 6, a first alternative confidence level of 8, and a second alternative confidence level of 10. The fourth character also contains a primary character value of 110, a first alternative character value of 52, and a second alternative character value of 56.

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev489_Vend
or">
    <V n="TYPE">Vendor</V>
    <V n="Position">0,0,0,0</V>
    <V n="STATUS">1</V>
    <C n="6,8,10" cr="0,0,0,0">83,49,53</C>
    <C n="6,8,10" cr="0,0,0,0">116,50,54</C>
    <C n="6,8,10" cr="0,0,0,0">105,51,55</C>
    <C n="6,8,10" cr="0,0,0,0">110,52,56</C>
</F>
```

Syntax

VBScript

```
oDCO.AddValue (Value as Long, Confidence as Long) as Boolean
```

C#

```
int AddValue(int nValue, int nConfidence)
```

Applies to

Character objects only.

Arguments

nValue

The ASCII code of the character.

nConfidence

A number between 0 (low) and 9 (high) representing the confidence level.

Returns

VBScript

Returns true if successful; returns false if unsuccessful.

C#

Returns 1 if successful; returns 0 if unsuccessful.

C# example

This example adds the ASCII character 50 with a confidence level of 3 to the character object:

```
m_oDCOChar.AddValue(50,3);
```

Parent topic: [DCO methods](#)

Related reference:

[DeleteValue method](#)

AddVariable method

The AddVariable method adds a variable and its value to an object, such as a batch, document, page, or field. You can use this method when you need to pass values to a custom action. For example, your organization might introduce a new security regulation that requires a new field for identification on certain page types.

Syntax

VBScript

```
oDCO.AddVariable (strName as String, newValue as Variant) as Boolean
```

C#

```
bool m_oDCO.AddVariable (string strName, object newValue)
```

Arguments

strName

The variable name.
newValue
The value of the variable.

Returns

Returns true if successful; returns false if unsuccessful.

Applies to

All objects.

Examples

VBScript

This example adds the *Company* variable with a value of IBM to the page object.

```
Dim page
Call objField.AddVariable ("Company", IBM)
factor = objField.Variable("Company")
```

C#

This example adds the variable *Company* and the value IBM to page TM000001 in the runtime batch.

```
TDCOLib.IDCO m_oChild = m_oDCO.FindChild("TM000001");
if (m_oChild != null
)
{
m_oChild.AddVariable("Company", "IBM");
}
```

The resulting XML code is shown in the following example:

```
<?xml-stylesheet type="text/xsl"
href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_..\..
\dco.xsl"?>
<B
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev490
_20100096.001">
  <V n="TYPE">APT</V>
  <V n="STATUS">0</V>
  <D
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev490
_20100096.001.01">
  <V n="TYPE">Invoice</V>
  <V n="STATUS">0</V>
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev490
_TM000001">
  <V n="TYPE">Main_Page</V>
  <V n="STATUS">0</V>
  <V n="Company">IBM</V>
```

Parent topic: [DCO methods](#)

Related reference:

[DeleteVariable method](#)

[set_Variable method](#)

[Variable property](#)

AddVariableFloat method

The `AddVariableFloat` method adds a variable of type `double` and its value to an object.

Syntax

VBScript

```
oDCO.AddVariableFloat (strName as String, fValue as Double) as Boolean
```

C#

```
bool AddVariableFloat(string strName, double fValue)
```

Applies to

All object types.

Arguments

`strName`

The variable name.

`fValue`

The value of the variable as a double.

Returns

Returns true if successful; returns false if unsuccessful.

Example

See [AddVariable method](#).

Parent topic: [DCO methods](#)

Related reference:
[set_Variable method](#)

AddVariableInt method

The `AddVariableInt` method adds an integer variable and its value to an object. You can use this method when you need to pass values to custom actions.

Syntax

VBScript

```
oDCO.AddVariableInt (strName as String, nValue as Integer) as Boolean.
```

C#

```
bool AddVariableInt(string strName, int nValue)
```

Applies to

All object types.

Arguments

strName

The variable name

nValue

The value of the variable as an integer. This method also accepts a double, but rounds the double to the nearest integer before the method assigns the value to the object.

Returns

Returns true if successful; returns false if unsuccessful.

Example

See [AddVariable method](#).

Parent topic: [DCO methods](#)

Related reference:

[set_Variable method](#)

AddVariableString method

The `AddVariableString` method adds a string variable and its value to an object. You can use this method when you need to pass values to custom actions.

Syntax

C#

```
bool AddVariableString(string strName, string strValue)
```

VBScript

```
oDCO.AddVariableString (strName as String, strValue as String) as Boolean
```

Applies to

All objects types.

Arguments

strName

The variable name.

strValue

The value of the variable as a string.

Returns

Returns true if successful; returns false if unsuccessful.

Example

See [AddVariable method](#).

Parent topic: [DCO methods](#)

Related reference:
[set_Variable method](#)

CheckIntegrity method

The `CheckIntegrity` method determines whether a batch conforms to the document integrity rules that are specified in the Setup DCO file.

This method is typically used after the [CreateDocuments method](#).

Syntax

VBScript

```
oDCO.CheckIntegrity (pLastChecked as Object ) as Long.
```

C#

```
int CheckIntegrity(out object pLastChecked)
```

Arguments

`pLastChecked`

Variable to contain the last object that is checked in case of an error.

Returns

One of the following values:

0

Passed

1

Has more child objects than allowed by the `max` attribute.

2

Has fewer child objects than required by the `min` attribute.

3

Invalid member. A child object is not of a type that is supported by the parent.

4

A child object is in the wrong position relative to other child objects as specified by the `pos` attribute.

Applies to

All object types except character objects. When applied to a batch or document, the method checks to the page level but not lower to a field or character object. When applied to a page, the method checks fields on that page.

VBScript example

This example creates documents in a batch and applies integrity rules. If an error is encountered, the code opens a message box that displays the type of error (1-4). The message box also displays the object and the ID that are the source of the error.

```
Dim DocInt
Dim LastChecked
Call oDCO.CreateDocuments
DocInt = oDCO.CheckIntegrity(LastChecked)
If DocInt > 0 Then
msgbox "Document Integrity problem = " & DocInt
msgbox "Detected at " & LastChecked.ObjectType & " type object ID " & LastChecked.ID
End If
```

C# example

This example populates the runtime and setup DCO objects, and checks the integrity of the runtime batch.

```
m_oDCO.Read("C:\\Datacap\\APT\\batches\\20100096.001\\Verify.xml");
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
Object pLastChecked = null;
int nRetVal = m_oDCO.CheckIntegrity(out pLastChecked);
```

Parent topic: [DCO methods](#)

Clear method

The `Clear` method removes all of the child objects of a runtime object, but does not remove the object itself. For example, you can use this method when a document contains corrupted or incorrect pages.

Syntax

VBScript

```
oDCO.Clear ( ) as Boolean.
```

C#

```
bool Clear()
```

Arguments

None.

Returns

Returns true if successful; returns false if unsuccessful.

Applies to

All object types.

VBScript example

This example instantiates a field-level object, uses `Clear ()` to remove any child objects, and then assigns a value to the `ID` property of the object.

```
Dim objFldTwo
Set objFldTwo = DCOSetup.GetNode(3,8)
objFldTwo.Clear
objFldTwo.ID = Tax2
```

Parent topic: [DCO methods](#)

Related reference:

[DeleteChild method](#)

CreateDocuments method

The `CreateDocuments` method uses the Setup DCO file to determine the document type that is associated with each page in a batch and creates the required document objects.

This method is normally called after the page identification task to assign pages to documents within the runtime hierarchy.

Syntax

VBScript

```
oDCO.CreateDocuments ( ) as Boolean.
```

C#

```
bool CreateDocuments()
```

Applies to

Batch objects only.

Arguments

None.

Returns

Returns true if successful; returns false if unsuccessful.

VBScript example

This example reads the Setup DCO file (BDOcs.xml), creates the document objects, and confirms that the batch conforms to the document integrity rules.

```
Call oDCO.ReadSetup ("c:\Datacap\BDOcs\dco_APT\BDOcs.xml")
Call objBatch.CreateDocuments
Call objBatch.CheckIntegrity
```

Parent topic: [DCO methods](#)

Related reference:

[CheckIntegrity method](#)

CreateFields method

The `CreateFields` method creates a runtime field object for each field that is specified within the corresponding page (or field) node in the document hierarchy.

For example, the `1040EZ` page node in the following document hierarchy has rules for three field objects. So, `CreateFields ()` creates three field objects (`Tax Year`, `SSN`, `Spouse SSN`).

```
<P type="1040EZ">
  (
    <V n="ID">0</V>
    <V n="TYPE">Page</V>
    <V n="STATUS">0</V>
    <V n="IMAGEFILE"></V>
    <V n="DATAFILE"></V>
    <V n="TEMPLATE IMAGE"></V>
    <V n="MIN_TYPES">1</V>
    <V n="MAX_TYPES">0</V>
    <V n="rules"><in><r
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev503_3"
rs="4" /><r
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev503_1"
rs="5" /><r
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev503_2"
rs="6" /></in></V>
    <F type="Tax Year" pos="0" min="0" max="0"/>      <!-- Rule 0 -->
    <F type="SSN" pos="0" min="0" max="0"/>          <!-- Rule 1 -->
    <F type="Spouse SSN" pos="0" min="0" max="0"/>   <!-- Rule 2 -->
  </P>
```

The child objects are used to hold the field data that is generated during the field recognition process. This method ensures that a data file is created for the current page when a batch is saved.

Syntax

VBScript

```
oDCO.CreateFields( ) as Boolean.
```

C#

```
bool CreateFields()
```

Arguments

None.

Returns

Returns true if successful; returns false if unsuccessful.

Applies to

Page objects or field objects.

Parent topic: [DCO methods](#)

Related reference:

[Write method](#)

DeleteChild method

The `DeleteChild` method removes the specified child object from the runtime DCO object.

Syntax

VBScript

```
oDCO.DeleteChild (nIndex subscript as Long ) as Boolean
```

C#

```
bool DeleteChild(int nIndex)
```

Applies to

Any object

Arguments

`nIndex`

Index of the child to delete, where 0 is the first child.

Returns

Returns true if successful; returns false if unsuccessful.

C# example

This example removes the first field object of the runtime page object.

```
m_oDCOPage.DeleteChild(0);
```

Parent topic: [DCO methods](#)

Related reference:

[AddChild method](#)

[Clear method](#)

DeleteValue method

The `DeleteValue` method deletes a data value and the corresponding confidence level value of a character object.

In the following example, the fourth character contains a primary confidence level value of 6, a first alternative confidence level of 8, and a second alternative confidence level of 10. The fourth character also contains a primary character value of 110, a first alternative character value of 52, and a second alternative character value of 56.

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev508_Vendor">
    <V n="TYPE">Vendor</V>
    <V n="Position">0,0,0,0</V>
    <V n="STATUS">1</V>
    <C n="6,8,10" cr="0,0,0,0">83,49,53</C>
    <C n="6,8,10" cr="0,0,0,0">116,50,54</C>
    <C n="6,8,10" cr="0,0,0,0">105,51,55</C>
```

```
<C n="6,8,10" cr="0,0,0,0">110,52,56</C>
```

```
</F>
```

Syntax

VBScript

```
oDCO.DeleteValue (nIndexValue as Long) as Boolean
```

C#

```
bool DeleteValue(int nIndexValue)
```

Applies to

Character objects only.

Arguments

nIndexValue

The index of the value to be deleted, where 0 is the primary value, 1 is the first alternative value, and so on.

Returns

Returns true if successful; returns false if unsuccessful.

C# example

This example deletes the first alternative character value and the first alternative confidence level value.

```
m_oDCOChar.DeleteValue(1);
```

Character XML before: `"\t<C cn="10,9" cr="0,0,0,0">83,99</C>\n"`

Character XML after: `"\t<C cn="10" cr="0,0,0,0">83</C>\n"`

Parent topic: [DCO methods](#)

Related reference:

[AddValue method](#)

DeleteVariable method

The `DeleteVariable` method removes a variable, including variables such as TYPE or STATUS that are installed with Datacap, and its value from a runtime DCO object.

Syntax

VBScript

```
oDCO.DeleteVariable (lpszName as String ) as Boolean
```

C#

```
bool DeleteVariable(string lpszName)
```

Applies to

All objects.

Arguments

lpszName
The name of the variable to delete (case sensitive).

Returns

Returns true if successful; returns false if unsuccessful.

Parent topic: [DCO methods](#)

Related reference:
[AddVariable method](#)

FindChild method

The FindChild method gets an interface to a child object that is referenced by ID (name). You can use this method when you need to get the interface to a child object for which you only know the name.

If the method references a field and the same ID is used within multiple page objects, the method returns the first matching field.

Syntax

VBScript

```
oDCO.FindChild (lpszName as String) as Boolean.
```

C#

```
TDCOLib.DCO.FindChild(string lpszName)
```

Arguments

lpszName
The ID (name) of the child object.

Returns

Returns an interface to the child object if successful; returns nothing if unsuccessful.

Applies to

All objects.

VBScript example

This example returns the interface to the Client_ID field, and displays the interface in a message box.

```
Dim objRTClFld  
Set objRTClFld=oDCO.FindChild("Client_ID")  
msgbox objRTClFld.ID
```

To retrieve a child object by index instead of ID, use [GetChild method](#)

C# example

This example returns an interface to the Vendor field:

```
m_oDCOField = m_oDCOPage.FindChild("Vendor");
```

Parent topic: [DCO methods](#)

Related reference:

[AddChild method](#)

[DeleteChild method](#)

[FindChildIndex method](#)

FindChildIndex method

The FindChildIndex method returns the index of the specified child object that is contained within a runtime DCO object. You can use this method when you know the name of the child object, and you need to pass the position of the child object to a variable.

Syntax

VBScript

```
oDCO.FindChildIndex (lpszName as String ) as Long
```

C#

```
int FindChildIndex(string lpszName)
```

Applies to

All objects except batch.

Arguments

lpszName

The ID (name) of the child object.

Returns

Returns the index of the specified child object, where 0 is the first child; returns -1 if the specified child is not found.

VBScript example

This example returns the index of the Client_ID field, and passes the index to the GetChild method.

```
Dim ClientFldVal  
ClientFldVal = objRTSurvey.FindChildIndex("Client_ID")  
Call objRTSurvey.GetChild(ClientFldVal)
```

C# example

This example returns the index of the Vendor field:

```
int nVendorIndex = m_oDCOPage.FindChildIndex("Vendor");
```

Parent topic: [DCO methods](#)

Related reference:

[FindChild method](#)

[GetChild method](#)

FindRouteChild method

The FindRouteChild method returns the interface to an object that is specified by using the path through the document hierarchy. You use this method to save or restore execution points during an action process.

The use of the path through a document hierarchy is shown in this example:

```
B/D(20100028.002.01)/P(TM000001)/F(Vendor)
```

Syntax

C#

```
TDCOLib.DCO FindRouteChild(string bszRoute)
```

Arguments

bszRoute

The route through the document hierarchy. (See the C# example.)

Returns

Returns an interface to the specified object; returns null if the route is invalid.

C# example

This example gets an interface to the Vendor field object on page TM000001:

```
m_oDCOField = m_oDCO.FindRouteChild("B/D(20100028.002.01)/P(TM000001)/F(Vendor)");
```

Parent topic: [DCO methods](#)

Related reference:

[GetRoute method](#)

FindVariable method

The FindVariable method returns the index of a variable by using the ID (name) of the variable. You can use this method when you need to pass the value of a variable to another variable or action.

Syntax

VBScript

```
oDCO FindVariable (lpszName as String ) Variables Index subscript as Long
```

C#

```
int FindVariable(string lpszName)
```

Arguments

lpszName
The variable ID (name).

Returns

Returns the index of the specified variable, where 0 is the first variable; Returns -1 if the specified variable is not found.

Applies to

All objects.

VBScript example

This example returns the index of the *Length* variable, and displays the index in a message box:

```
Dim VarLength
varLength = oDCO.FindVariable("Length")
if varLength > 0 then
    msgbox oDCO.Variable(varLength)
end if
```

Parent topic: [DCO methods](#)

Related reference:

[get_Variable method](#)

[set_Variable method](#)

[Variable property](#)

get_AltConfidenceString method

The `get_AltConfidenceString` method gets the confidence level of each character in the referenced field. You can use this method when you need to pass the confidence level of a string to a variable or action.

.NET only: For VBScript, use the [AltConfidenceString property](#) instead.

The confidence level of each character is a digit from 0 (lowest confidence) to 9 (highest confidence). If the confidence level for a four-character field is 9999, each of the 4 characters has a confidence level of 9. The XML code adds a value of 1 to the confidence level so that the range in XML is 1-10. In the following example XML setup node for the field, `Vendor`, the confidence level for the primary character is 6. The confidence level for the first alternative character is 8, and the confidence level for the second alternative is 10.

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev527_Vendor">
    <V n="TYPE">Vendor</V>
    <V n="Position">0,0,0,0</V>
    <V n="STATUS">1</V>
    <C cn="6,8,10" cr="0,0,0,0">83,49,53</C>
    <C cn="6,8,10" cr="0,0,0,0">116,50,54</C>
    <C cn="6,8,10" cr="0,0,0,0">105,51,55</C>
```

```
</F> <C cn="6,8,10" cr="0,0,0,0">110,52,56</C>
```

Syntax

C#

```
string get_AltConfidenceString(int nIndex)
```

Applies to

Field objects only.

Arguments

nIndex

0 specifies the primary value. Where the field contains multiple recognition, voting, or multiple-pass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on. (See the C# example.)

Returns

String containing the confidence level of each character. The returned digits are always one less than the digits that are stored in the page XML file. (See the C# example.)

C# example

The following example gets the confidence string for the second alternative.

```
string strAltConf = m_oVendorField.get_AltConfidenceString(2)
```

Referencing the XML code example for the Vendor field, the return value in this case is 9999, because the returned value is always the stored value minus 1.

Parent topic: [DCO methods](#)

Related reference:

[ConfidenceString property](#)
[get_CharConfidence method](#)
[set_AltConfidenceString](#)
[set_CharConfidence method](#)

get_AltText method

The `get_AltText` method gets the primary or alternative character data that is associated with a field.

.NET only: For VBScript, use [AltText property](#) instead.

Primary and alternative values are stored in ASCII format in the page XML file, as shown in the following example:

```
<F  
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev528_Vend  
or">  
    <V n="TYPE">Vendor</V>  
    <V n="Position">0,0,0,0</V>  
    <V n="STATUS">1</V>  
    <C n="6,8,10" cr="0,0,0,0">83,49,68</C>
```

```
<C n="6,8,10" cr="0,0,0,0">116,50,97</C>
<C n="6,8,10" cr="0,0,0,0">105,51,116</C>
<C n="6,8,10" cr="0,0,0,0">110,52,97</C>
</F>
```

For the fourth character, the primary text value is 110. The first alternative text value is 52, and the second alternative text value is 97.

Syntax

C#

```
string get_AltText(int nIndex)
```

Applies to

Field objects only.

Arguments

nIndex

0 specifies the primary value. When the field contains multiple recognition, voting, or multiple-pass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

Returns

String containing the alternative text.

Example

This example gets the second alternative text value from the Vendor field. Using the page XML example, the returned string is 68,97,116,97. The root of the search in this case is the page DCO.

```
string sAltText = m_oDCOPage.FindChild("Vendor").get_AltText(2);
```

Parent topic: [DCO methods](#)

Related reference:

[get_CharValue method](#)

[set_AltText method](#)

[Text property](#)

get_CharConfidence method

The `get_CharConfidence` method gets the primary or alternative character data that is associated with a field. You can use this method when you need to pass the character data value to a variable or action.

.NET only: For VBScript, use [CharConfidence property](#) instead.

This method gets the confidence level (cn) of the primary character value or alternative value. Within the XML file, the confidence level is a digit from 1 (lowest confidence) to 10 (highest confidence). In the following example XML setup node for the field, `Vendor`, the character primary confidence level is 6. The first alternative confidence level is 8, and the second alternative confidence level is 10.

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev529_Vendor">
```

```
<V n="TYPE">Vendor</V>
<V n="Position">0,0,0,0</V>
<V n="STATUS">1</V>
<C cn="6,8,10" cr="0,0,0,0">83,49,53</C>
<C cn="6,8,10" cr="0,0,0,0">116,50,54</C>
<C cn="6,8,10" cr="0,0,0,0">105,51,55</C>
<C cn="6,8,4" cr="0,0,0,0">110,52,56</C>
</F>
```

Syntax

C#

```
int get_CharConfidence(int nIndex)
```

Applies to

Character objects only,

Arguments

nIndex

0 specifies the primary value. When the field contains multiple recognition, voting, or multiple-pass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

Returns

Integer representing the confidence level.

- `get_CharConfidence` returns the value as it is stored in the XML file. The field-level functions, such as [get_AltConfidenceString method](#), return the internal representation, which is the stored value minus 1.

C# example

This example gets the confidence level of second alternative value of the fourth character (index = 3) in the Vendor field. By using the example XML setup node, the returned value is 4. The root of the search in this case is the page DCO.

```
nConfLevel = m_oDCOPage.FindChild("Vendor").GetChild(3).get_CharConfidence(2);
```

Parent topic: [DCO methods](#)

Related reference:

[get_AltText method](#)

[get_CharValue method](#)

[set_CharConfidence method](#)

get_CharValue method

The `get_CharValue` method gets the ASCII data value of the primary character or alternative character. You can use this method when you need to pass the character data value to a variable or action.

.NET only: For VBScript, use [CharValue property](#) instead.

In the following example XML setup node for the field, `Vendor`, the fourth character has an ASCII data value of 110 for the primary character. Also, the ASCII data value is 52 for the first alternative character and 56 for the second alternative character.

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev530_Vendor">
    <V n="TYPE">Vendor</V>
    <V n="Position">0,0,0,0</V>
    <V n="STATUS">1</V>
    <C cn="6,8,10" cr="0,0,0,0">83,49,53</C>
    <C cn="6,8,10" cr="0,0,0,0">116,50,54</C>
    <C cn="6,8,10" cr="0,0,0,0">105,51,55</C>
    <C cn="6,8,10" cr="0,0,0,0">110,52,56</C>
</F>
```

Syntax

C#

```
int get_CharValue(int nIndex)
```

Applies to

Character objects only.

Arguments

nIndex

0 specifies the primary value. Where the field contains multiple recognition, voting, or multi-pass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

Returns

Integer representing the ASCII character value.

- To get the converted text of the entire field, use [get_AltText method](#).

C# example

This example gets the second alternative value of the fourth character (index = 3) in the Vendor field. By using the example XML setup node, the returned value is 56. The root of the search in this case is the page DCO.

```
nCharValue = m_oDCOPage.FindChild("Vendor").GetChild(3).get_CharValue(2);
```

Parent topic: [DCO methods](#)

Related reference:

[get_CharConfidence method](#)

[set_CharValue method](#)

get_OMRValue method

The `get_OMRValue` method gets the positions within the optical mark recognition (OMR) field of the check boxes that are selected. You can use this method to pass the positions to variables or actions, such as those that are used in verification functions to confirm that check boxes are selected.

Each character in an OMR field is assigned the value 0 (ASCII 48 = not selected) or 1 (ASCII 49 = selected), and the resulting binary string is expressed as a decimal. The following XML example is 000100, which equals 4 in decimal. So, the return value is 4.

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev532_1a">
  <V n="TYPE">1a</V>
  <V n="Position">120,1025,853,1169</V>
  <V n="STATUS">0</V>
  <V n="DensityString">BAAVAA</V>
  <C cn="10" cr="519,1132,559,1170">48</C> <!--48 indicates option not selected-->
  <C cn="10" cr="563,1132,603,1170">48</C>
  <C cn="10" cr="607,1132,647,1170">48</C>
  <C cn="10" cr="652,1132,692,1170">49</C> <!--49 indicates option selected-->
  <C cn="10" cr="697,1132,737,1170">48</C>
  <C cn="10" cr="763,1132,803,1170">48</C>
</F>
```

Syntax

C#

```
int get_OMRValue(int nIndex)
```

Arguments

nIndex

0 specifies the primary value. When the field contains multiple recognition, voting, or multi-pass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

Returns

The positions of the selected OMR check boxes (See the XML example.)

Applies to

OMR fields only.

Parent topic: [DCO methods](#)

Related reference:

[get_CharValue method](#)

[set_OMRValue](#)

get_Variable method

The `get_Variable` method gets the value of a referenced variable by using the ID (name) of the variable. You can use this method to pass the value of a named variable to an action for further processing.

.NET only: For VBScript, use [Variable property](#) instead.

If the specified variable does not exist in the runtime object, but does exist in the associated setup node object, the `get_Variable` method returns the value from the setup node.

A number of variables are defined and installed with Datacap. For a listing of these variables, see the *Standard Variable Reference* in IBM® Knowledge Center for Datacap.

Syntax

C#

```
string get_Variable(string lpszName)
```

Arguments

lpszName

The name of the variable of which you need the value.

Applies to

All objects.

C# example

This example gets the value of the *Company* variable.

```
string CompanyName = m_oDCO.get_Variable("Company")
```

Parent topic: [DCO methods](#)

Related reference:

[GetVariableName method](#)

[GetVariableValue method](#)

[set_Variable method](#)

GetChild method

The `GetChild` method gets a referenced child object by using the index of the child object. You can use this method to retrieve a field on a page when you know the position, but not the name.

Syntax

VBScript

```
oDCO.GetChild (nIndex as Long ) as Object
```

C#

```
TDCOLib.DCO.GetChild(int nIndex)
```

Arguments

nIndex

The index of the child, where 0 is the first child

Returns

Returns the child object if successful; returns nothing otherwise.

Applies to

All objects.

VBScript example

This example gets the name of the first four fields on a page, and displays each name in a message box.

```
Dim objViewField(4)
Dim I
i = 0
Do While i < 4
    Set objViewField(i) = oDCO.GetChild(i)
    msgbox objViewField(i).Text
    i = i + 1
Loop
```

C# example

This example returns the first field object on the page:

```
m_oDCOField = m_oDCOPage.FindChild(0);
```

- You can retrieve a child object by ID instead of by index by using the [FindChild method](#).
- You can create a loop by using the GetChild method with the [NumOfChildren method](#) to get all of the child objects that are associated with an object.

Parent topic: [DCO methods](#)

Related reference:

[FindChildIndex method](#)

GetLastError method

The GetLastError method retrieves the text of the last error that is encountered during a DCO read operation or write operation, and clears the error.

Before you call the GetLastError method, you can check whether an error string exists by calling the [IsError method](#).

Syntax

VBScript

```
objRTBatch.GetLastError as Boolean
```

C#

```
string GetLastError()
```

Arguments

None.

Returns

Returns true if successful; returns false if unsuccessful.

Applies to

Batch objects only.

VBScript example

```
objBatch.BatchDir = c:\Datacap\MQSW\Batches
bStatus = objBatch.Write rulerunner.xml
If bStatus <> True Then
    msgbox objBatch.GetLastError
End if
```

Parent topic: [DCO methods](#)

Related reference:

[Read method](#)

[Write method](#)

GetPosition method

The `GetPosition` method gets the position of a field or character on a page. You can use this method for fields or characters that are located at identical positions on multiple pages in a batch.

- For field objects, the position is defined by the *Position* variable of the field.

```
<V n="Position">0,0,0,0</V>
```

- For character objects, the position is defined by the `cr` attribute of the character.

```
<C cn="10" cr="0,0,0,0">83</C>
```

Syntax

VBScript

```
oDCO.GetPosition(pnLeft as Long, pnTop as Long, nRight as Long, pnBottom as Long) as Boolean
```

C#

```
bool GetPosition(out object pnLeft, out object pnTop, out object pnRight, out object pnBottom)
```

Arguments

Argument	Description
pnLeft	Variable for the distance from the left side of the page to the left edge of the object (in pixels)
pnTop	Variable for the distance from the top of the page to top edge of the object (in pixels)
pnRight	Variable for the distance from the left side of the page to right edge of the object (in pixels)
pnBottom	Variable for the distance from the top of the page to bottom edge of the object (in pixels)

Returns

Returns true if successful; returns false if unsuccessful.

Applies to

Field and character objects only.

VBScript example

This example gets a field's distance in pixels from the left, top, right, and bottom of a page, and displays the values in a message box.

```
Dim L,T,R,B
oDCO.GetPosition "L,T,R,B"
msgbox "The field's position is " L & ", " & T & ", " & R & ", " & B
```

C# example

This example finds the Vendor field on a page, and gets the field's distance in pixels from the left, top, right, and bottom of a page.

```
m_oDCOField = m_oDCOPage.FindChild("Vendor");
object pLeft, pTop, pRight, pBottom;
m_oDCOField.GetPosition(out pLeft, out pTop, out pRight, out pBottom);
```

Parent topic: [DCO methods](#)

Related reference:
[SetPosition method](#)

GetRoute method

The GetRoute method gets the path through the document hierarchy to an object.

Use the argument to specify whether you want the path through the Runtime DCO (`True`) or the Setup DCO (`False`).

Syntax

```
C#
string GetRoute(bool bRuntime)
```

Arguments

bRuntime
True for the Runtime DCO path; False for the Setup DCO path

Returns

String with the path.

Applies to

Any object type.

C# example

In the first example, `GetRoute ()` returns the path through the Runtime DCO to the Vendor field on page TM000001.

```
m_oDCO.Read("C:\\Datacap\\APT\\batches\\20100096.001\\Verify.xml");
m_oDCOPage = m_oDCO.FindChild("TM000001");
m_oDCOField = m_oDCOPage.FindChild("Vendor");
string strRoute = m_oDCOField.GetRoute(true);
```

The returned string has the following format: B/D(20100096.001.01)/P(TM000001)/F(Vendor).

In the next example, `GetRoute()` returns the path through the Setup DCO to the Details field.

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");  
m_oDCOField = m_oDCO.FindChild("Details");  
string returnString = m_oDCOField.GetRoute(false);
```

The returned string has the following format: B/D[Invoice]/P[Main_Page]/F[Details].

Parent topic: [DCO methods](#)

Related reference:

[FindRouteChild method](#)

[IsRoute method](#)

GetVariableName method

The `GetVariableName` method gets the name (ID) of a variable from the index of the variable. You can use this method when you know the position of a variable within an object, but you do not know the name of the variable.

Syntax

VBScript

```
oDCO.VariableName(nIndex subscript as Long) as String
```

C#

```
string GetVariableName(int nIndex)
```

Applies to

All objects.

Arguments

nIndex

The index of the variable.

Returns

Returns the name of the variable; returns an empty string if a variable with the specified index does not exist.

VBScript example

This example returns the name of the third variable in a field object.

```
Dim VarID  
VarID = objField.VariableName(2)
```

Parent topic: [DCO methods](#)

Related reference:

[FindVariable method](#)

[NumOfVars method](#)

GetVariableValue method

The GetVariableValue method gets the value of a referenced variable by using the index of the variable.

Syntax

VBScript

```
oDCO.VariableName(nIndex subscript as Long) as Variant
```

C#

```
object GetVariableValue(int nIndex)
```

Applies to

All objects.

Arguments

nIndex

The index of the variable, the value of which you want to get, where 0 is the first variable.

Returns

Returns an object that contains the value of the variable; returns nothing if there is an error.

VBScript example

This example gets the value of the third variable in a field object, and displays the value of the variable in a message box.

```
Dim VarVal  
VarVal = objField.VariableValue(2)  
msgbox VarVal
```

Tip: To get the value of a referenced variable by using the ID (name) of the variable, use the [Variable property](#) property.

C# example

This example gets the value of the 10th variable that is defined on page 1 of the runtime batch.

```
object oVarValue = m_oDCO.FindChild("TM000001").GetVariableValue(9);
```

Tip: To get the value of a referenced variable by using the ID (name) of the variable, use the [get_Variable method](#) method.

Parent topic: [DCO methods](#)

Related reference:
[FindVariable method](#)

IsError method

The IsError method indicates whether an error occurred during a prior Read operation or Write operation.

If `IsError` returns true, use the [GetLastError method](#) to get the associated error message. The `IsError` flag is cleared upon calling `GetLastError`.

Syntax

VBScript

```
oDCO.IsError as Boolean
```

C#

```
bool IsError()
```

Applies to

Batch object only.

Arguments

None.

Returns

Returns true if the `IsError` flag is set; Returns false if the `IsError` flag is not set.

VBScript example

This example checks if there are any errors for the Rulerunner task. If an error is found, it displays the error to a message box.

```
objBatch.BatchDir = "c:\Datacap\MQSW\Batches"  
' clear any old errors  
sOldError = objBatch.GetLastError  
Call objBatch.Write ("rulerunner.xml")  
If objBatch.IsError <> 0 Then  
    msgbox objBatch.GetLastError  
End if
```

Parent topic: [DCO methods](#)

Related reference:

[Read method](#)

[Write method](#)

IsRoute method

The `IsRoute` method indicates whether the specified path is the valid path through the runtime hierarchy for an object.

Syntax

C#

```
bool IsRoute(string lpszRoute)
```

Arguments

lpszRoute

The path through the runtime hierarchy. (See the C# example.)

Returns

Returns true if the specified path matches the path for this object; returns false if the specified path does not match the path for this object.

C# example

The following example gets an interface to the Vendor field and then uses `GetRoute` to confirm the path.

```
m_oDCOField = m_oDCO.FindChild("TM000001").FindChild("Vendor");  
bool bRouteOK = m_oDCOField.IsRoute("B/D(20100028.002.01)/P(TM000001)/F(Vendor)");
```

Parent topic: [DCO methods](#)

Related reference:

[FindRouteChild method](#)

[GetRoute method](#)

IsValid method

The `IsValid` method confirms that the interface of a DCO object is valid and is connected to an actual object.

Syntax

VBScript

```
oDCO.IsValid as Boolean
```

C#

```
bool IsValid()
```

Arguments

None.

Returns

Returns true if the interface is valid; returns false if the interface is invalid.

Applies to

Field objects only.

C# example

The following call returns `False`:

```
TDCOLib.IDCO m_oNullDCOObject = new TDCOLib.DCOClass();  
bResult = m_oNullDCOObject.IsValid();
```

Parent topic: [DCO methods](#)

MoveChild method

The `MoveChild` method moves a child of a runtime DCO object to a different index location. You can use this method to reorganize or correct batches, documents, pages, or fields.

Description

For example, `MoveChild(0, 2)` moves the object at position 0 to position 2. When you use this method, `MoveChild` reindexes the other child objects, as needed.

Table 1. Example implementation of `MoveChild` method

Index	Value		Index	Value
0	One	MoveChild(0, 2)	0	Two
1	Two		1	Three
2	Three		2	One
Before			After	

Syntax

VBScript

```
oDCO.MoveChild(nOldIndex as Long, nNewIndex as Long) as Boolean
```

C#

```
bool MoveChild(int nOldIndex, int nNewIndex)
```

Arguments

`nOldIndex`

Index value of the current position, of the child object, where 0 is the first child

`nNewIndex`

Index value of the new position of the child object

Returns

True if successful; False otherwise

Applies to

All objects

C# Example

This example moves the first field on page TM000001 to the third position.

```
m_oDCO.FindChild("TM000001").MoveChild(0, 2);
```

See also

[FindChild method](#), [MoveIn method](#)

Parent topic: [DCO methods](#)

MoveIn method

The `MoveIn` method moves the specified runtime DCO object from the current parent to a different parent.

Syntax

VBScript

```
oDCO.MoveIn(pNewParent as Object, nIndex as Long) as Boolean.
```

C#

```
bool MoveIn(object pNewParent, int nIndex)
```

Applies to

All objects

Arguments

`pNewParent`

A valid runtime DCO object as the new parent

`nIndex`

The index value for the child in the new parent. An index value of `-1` places the child at the end.

Returns

True if successful; False otherwise

C# Example

The following example moves the Vendor field from page `TM000002` to the end of page `TM000001`.

```
m_oDCOField = m_oDCO.FindChild("TM000002").FindChild("Vendor");  
m_oDCOPage = m_oDCO.FindChild("TM000001");  
m_oDCOField.MoveIn(m_oDCOPage, -1);
```

See also

[MoveChild method](#)

Parent topic: [DCO methods](#)

NumOfChildren method

The `NumOfChildren` method returns the number of child objects that is associated with the runtime DCO object.

Syntax

VBScript

```
oDCO.NumOfChildren as Long
```

C#

```
int NumOfChildren()
```

Arguments

None

Returns

The number of child objects that is associated with this object.

Applies to

All objects

VBScript Example

```
Dim CntPgChildren  
CntPgChildren = oDCO.NumOfChildren  
msgbox "Page 1 has " & CntPgChildren & " children"
```

C# Example

This example returns the number of fields on page TM000001.

```
int nNumFields = m_oDCO.FindChild("TM000001").NumOfChildren();
```

See also

[GetChild method](#)

Parent topic: [DCO methods](#)

NumOfVars method

The `NumOfVars` method returns the number of variables that is associated with the runtime DCO object.

Syntax

VBScript

```
oDCO.NumOfVars as Long
```

C#

```
int NumOfVars()
```

Arguments

None

Returns

A count of the variables of the object

Applies to

All objects

Example

This example returns the number of variables that is associated with the Vendor field.

```
int nNumVars = m_oDCO.FindChild("TM000001").FindChild("Vendor").NumOfChildren();
```

See also

[GetVariableValue method](#), [GetVariableName method](#)

Parent topic: [DCO methods](#)

ObjectType method

The `ObjectType` method returns a numeric value that indicates the object type.

Syntax

VBScript

```
oDCO.ObjectType as Long
```

C#

```
int ObjectType ()
```

Arguments

None.

Returns

A value that indicates the object type (0=batch, 1=document, 2=page, 3=field, 4=character).

Applies to

All objects

Example

VBScript

```
If oDCO.ObjectType <> 2 Then  
msgbox "The object is not a page!"  
End if
```

Parent topic: [DCO methods](#)

Parent

The `Parent` method returns the parent of the runtime DCO object.

Syntax

VBScript

```
oDCO.Parent as oDCO Object
```

C#

```
TDCOLib.DCO Parent()
```

Arguments

None

Returns

The parent object, or nothing (null) if the method is applied to a batch object.

Applies to

All objects except batch

Example

VBScript

```
If oField.Parent = Nothing Then  
msgbox "Cannot access the page!"  
End if
```

Parent topic: [DCO methods](#)

Read method

The `Read` method reads the runtime information from a Runtime DCO file or a page data file, and writes the information into the DCO object.

This method can read a Runtime DCO file (for example, `Verify.xml`), or a page data file (for example, `tm000001.xml`) of an application. Because new DCO objects are created as batch objects by default, you must first create a child object of type `page` before using the `Read` method to directly read the page data file. Runtime information files are located in the `batches` folder of an application, and include the data that are denoted by the following properties:

```
|--m_oDCO  
|  |--BatchDir(string)  
|  |--BatchPriority(int)  
|  |--ConfidenceString(string)  
|  |--ID(string)  
|  |--ImageName(string)  
|  |--Options(string)  
|  |--Status(int)
```

```
|--Text (string)
|--Type (string)
|--XML (string)
|--AltConfidenceString (string) *
|--AltText (string) *
|--CharConfidence (int) *
|--CharValue (int) *
|--OMRValue (int) *
|--Variable (string) *
```

* These extended properties are accessible directly from VBScript or through the corresponding get method or set method in .NET.

Syntax

VBScript

```
oDCO.Read (FilePath as String) as Boolean
```

C#

```
bool Read(string lpszFileName)
```

Applies to

Batch or page objects only.

Arguments

`lpszFileName`

The full path and name of the file to read.

Returns

Returns true if successful; returns false if unsuccessful.

Example

The first example reads the Runtime DCO file into an existing batch level DCO object:

```
m_oDCOBatch.Read("C:\\Datacap\\APT\\batches\\20100096.001\\Verify.xml");
```

The second example creates a runtime DCO object with a page data file only:

```
TDCOLib.DCO m_oDCO = new TDCOLib.DCO();
TDCOLib.DCO m_oDCOPage = new TDCOLib.DCO();
m_oDCOPage = m_oDCO.AddChild(2, "NewChildPage", -1);
m_oDCOPage.Read("C:\\Datacap\\APT\\batches\\20100096.001\\tm000001.xml");
```

If there is an error, the [IsError method](#) returns true and the [GetLastError method](#) returns the last error message.

Parent topic: [DCO methods](#)

Related reference:

[CreateFields method](#)

[FindChild method](#)

[ReadSetup method](#)

[Write method](#)

ReadSetup method

The ReadSetup method reads the document hierarchy setup information from the specified Setup DCO file and writes the information into the Setup object and SetupNode object.

This sample file shows properties that the ReadSetup method can process.

```
|--Setup
  |--Path (string)
  |--DictionaryName*
  |--Value*
  |--Word*

|--SetupNode
  |--MaxNumOfChildren (int)
  |--MinNumOfChildren (int)
  |--Name (string)
  |--ObjectType (int)
  |--RuleChildName*
  |--RuleMaxNumber*
  |--RuleMinNumber*
  |--RuleObjectType*
  |--RulePosition*
  |--Variable*
  |--VariableName*
  |--VariableValue*
```

* Extended properties that are accessible directly from VBScript or through the corresponding get method or set method in .NET.

Syntax

VBScript

```
oDCO.ReadSetup (FileName as String) as Boolean
```

C#

```
bool IDCO.ReadSetup(string lpszFileName)
```

Arguments

lpszFileName

Full path and name for the Setup DCO XML file. (See the following example.)

Returns

Returns true if successful; returns false if unsuccessful.

C# example

This example populates the Setup object and SetupNode object from APT.XML, changes the name of the first dictionary to Datacap, and writes the objects back to the Setup DCO.

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
m_oDCO.SetupObject().set_DictionaryName(0, "Datacap");
m_oDCO.WriteSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
```

Parent topic: [DCO methods](#)

Related reference:

[SetupNode method](#)

[SetupObject method](#)

[WriteSetup method](#)

SetPosition method

The SetPosition method sets the position of the runtime field or character object on the page. This method is useful for making corrections to zonal recognition on a page.

- For field objects, the position is defined by the *Position* variable of the field.

```
<V n="Position">0,0,0,0</V>
```

- For character objects, the position is defined by the `cr` attribute of the character.

```
<C cn="10" cr="0,0,0,0">83</C>
```

Syntax

VBScript

```
oDCO.SetPosition (nLeft as Long, nTop as Long, nRight as Long,  
nBottom as Long) as Boolean
```

C#

```
bool SetPosition(int nLeft, int nTop, int nRight, int nBottom)
```

Applies to

Field or character objects.

Arguments

`nLeft`

Distance from the left side of the page to the left edge of the object (in pixels)

`nTop`

Distance from the top of the page to the top edge of the object (in pixels)

`nRight`

Distance from the left side of the page to the right edge of the object (in pixels)

`nBottom`

Distance from the top of the page to the bottom edge of the object (in pixels)

Returns

Returns true if successful; returns false if unsuccessful.

Example

VBScript

```
Call oDCO.SetPosition (115, 450, 138, 400)
```

Parent topic: [DCO methods](#)

Related reference:[GetPosition method](#)

SetupNode method

The `SetupNode` method accesses the `SetupNode` object that is associated with the current object. You can use this method to get the name of the object in the document hierarchy that corresponds to the object in the runtime batch hierarchy.

Syntax

VBScript

```
oDCO.SetupNode as Object
```

C#

```
TDCOLib.DCOSetupNode SetupNode()
```

Arguments

None.

Returns

The setup object that corresponds to this runtime object. If you created this object dynamically at run time, and the object is not based on a `SetupNode`, then the method returns nothing (null).

Applies to

All objects.

C# example

This example populates the batch level `Setup` object and `SetupNode` object from the Setup DCO file and points `m_oDCOSetupNode` to the `SetupNode` object of the `Vendor` field.

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");  
TDCOLib.DCOSetupNode m_oDCOSetupNode =  
m_oDCO.FindChild("TM000001").FindChild("Vendor").SetupNode();
```

Parent topic: [DCO methods](#)**Related reference:**[ReadSetup method](#)[SetupObject method](#)[WriteSetup method](#)

SetupObject method

The `SetupObject` method returns the `Setup` object that is associated with the current batch-level runtime DCO object. You can use this method when a batch contains unidentified pages and you need to modify the runtime batch hierarchy.

Syntax

VBScript

```
oDCO.SetupObject as Object
```

C#

```
TDCOLib.DCOSetup SetupObject()
```

Applies to

Batch objects only.

Arguments

None.

Returns

The setup object.

C# example

This example populates the batch level Setup and SetupNode objects from the Setup DCO file and points `m_oDCOSetup` to the Setup object:

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");  
TDCOLib.DCOSetup m_oDCOSetup = m_oDCO.SetupObject();
```

For a complete example, see [ReadSetup](#).

Parent topic: [DCO methods](#)

Related reference:

[WriteSetup method](#)

set_AltConfidenceString

The `set_AltConfidenceString` method sets the confidence level of each character in the referenced field.

Description

Important: .NET only. For VBScript, use the [AltConfidenceString](#) property instead.

The confidence level of each character is a digit from 0 (lowest confidence) to 9 (highest confidence). For example, if the confidence level for a four-character field is 9999, each of the 4 characters has a confidence level of 9.

```
<F  
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev575_Vendor">  
  <V n="TYPE">Vendor</V>  
  <V n="Position">0,0,0,0</V>  
  <V n="STATUS">1</V>  
  <C cn="6,8,10" cr=0,0,0,0">83,49,53</C>  
  <C cn="6,8,10" cr=0,0,0,0">116,50,54</C>  
  <C cn="6,8,10" cr=0,0,0,0">105,51,55</C>
```

```
<C cn="6,8,10" cr=0,0,0,0">110,52,56</C>
</F>
```

In the sample XML file, the primary confidence level is 6. The first alternative confidence level is 8, and the second alternative confidence level is 10. The value that is stored is always the value set + 1.

Syntax

C#

```
void set_AltConfidenceString(int nIndex, string pVal)
```

Applies to

Field objects only

Arguments

nIndex

0 specifies the primary value. Where the field contains multiple recognition, voting, or multipass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on. (See the following example).

pVal

A string that contains the confidence level of each character (for example, "9999"). The digits that are stored in the page XML file are always one greater than the digits you specify in pVal (See the following example).

Comments

- To set or get the primary confidence level, you can also use the [ConfidenceString](#) property.
- To set or get the confidence level on individual characters, use [set_CharConfidence](#) or [get_CharConfidence](#).

Example

C#

The following example sets the confidence string for the second alternative.

```
m_oVendorField.set_AltConfidenceString(2, "9999")
```

The value 9999 is stored as 10,10,10,10 because the value that is stored is always the value set + 1.

See also

[get_AltConfidenceString](#), [set_AltText](#)

Parent topic: [DCO methods](#)

set_AltText method

The `set_AltText` method sets the primary character data or alternative character data that is associated with a field. You can use this method for multi-pass verification tasks and double-blind data entry tasks.

.NET restriction: For VBScript, use the [AltText](#) property instead.

The `set_AltText` method sets primary and alternative values that are stored in the page XML file in ASCII format, as shown in the following example:

```
F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev576_Vendor">
  <V n="TYPE">Vendor</V>
  <V n="Position">0,0,0,0</V>
  <V n="STATUS">1</V>
  <C cn="6,8,10" cr="0,0,0,0">83,49,68</C>
  <C cn="6,8,10" cr="0,0,0,0">116,50,97</C>
  <C cn="6,8,10" cr="0,0,0,0">105,51,116</C>
  <C cn="6,8,10" cr="0,0,0,0">110,52,97</C>
</F>
```

In the sample XML file, the primary text value for the fourth character is 110. The first alternative text value is 52, and the second alternative text value is 97.

Syntax

C#

```
void set_AltText(int nIndex, string pVal)
```

Applies to

Field objects only.

Arguments

`nIndex`

0 specifies the primary value. Where the field contains multiple recognition, voting, or multi-pass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

`pVal`

String containing the alternative text.

C# example

This example sets the alternative value 1 in the field object.

```
m_oField.set_AltText(1, "Hello");
```

- Use the `set_AltConfidenceString` method to specify a confidence level other than the default value of 9 (high confidence).
- Use the `Text` property to set the primary field value.
- Use the `set_CharValue` method to set individual characters.

Parent topic: [DCO methods](#)

Related reference:

[get_AltText method](#)

[set_AltConfidenceString](#)

[Text property](#)

[set_CharValue method](#)

set_CharConfidence method

The `set_CharConfidence` method sets the confidence level for the primary value or of the alternative value of a character in a field.

.NET Restriction: For VBScript, use the [CharConfidence](#) property instead.

The confidence level is determined by the `cn` attribute. Within the page XML file, the confidence level is a digit from 1 (lowest confidence) to 10 (highest confidence). In the following XML sample, the primary confidence level of the fourth character is 6. The first alternative confidence level is 8, and the second alternative confidence level is 4.

```
F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev577_Vendor">
  <V n="TYPE">Vendor</V>
  <V n="Position">0,0,0,0</V>
  <V n="STATUS">1</V>
  <C cn="6,8,10" cr="0,0,0,0">83,49,53</C>
  <C cn="6,8,10" cr="0,0,0,0">116,50,54</C>
  <C cn="6,8,10" cr="0,0,0,0">105,51,55</C>
  <C cn="6,8,4" cr="0,0,0,0">110,52,56</C>
</F>
```

Syntax

C#

```
void set_CharConfidence(int nIndex, int pVal)
```

Arguments

`nIndex`

0 specifies the primary value. Where the field contains multiple recognition, voting, or multipass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

`pVal`

The confidence value (1-10), as stored in the XML file.

This method sets the value as it is stored in the XML file (1-10). For field-level functions such as [set_AltConfidenceString](#), you specify the internal representation, which is the stored value minus 1.

C# example

This example sets the confidence level of second alternative value of the fourth character (index = 3) in the `Vendor` field. The root of the search in this case is the page DCO.

```
m_oDCOPage.FindChild("Vendor").GetChild(3).set_CharConfidence(2, 4);
```

Parent topic: [DCO methods](#)

Related reference:

[get_CharConfidence method](#)

set_CharValue method

The `set_CharValue` method sets the ASCII data value of the primary character or alternative character.

.NET Restriction: For VBScript, use the [CharValue](#) instead.

To set the text of the entire field, use the [set_AltText](#). In the following sample XML file, the primary value for the fourth character is 110. The first alternative value is 52, and the second alternative value is 56.

```
F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev578_Vendor">
  <V n="TYPE">Vendor</V>
  <V n="Position">0,0,0,0</V>
  <V n="STATUS">1</V>
  <C cn="6,8,10" cr="0,0,0,0">83,49,53</C>
  <C cn="6,8,10" cr="0,0,0,0">116,50,54</C>
  <C cn="6,8,10" cr="0,0,0,0">105,51,55</C>
  <C cn="6,8,10" cr="0,0,0,0">110,52,56</C>
</F>
```

Syntax

C#

```
void set_CharValue(int nIndex, int pVal)
```

Applies to

Character objects only.

Arguments

nIndex

0 specifies the primary value. Where the field contains multiple recognition, voting, or multipass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

Returns

Nothing.

C# example

This example sets the second alternative value of the fourth character (index = 3) in the `Vendor` field. The root of the search in this case is the page DCO.

```
m_oDCOPage.FindChild("Vendor").GetChild(3).set_CharValue(2, 56);
```

Parent topic: [DCO methods](#)

Related reference:

[get_CharValue method](#)

[set_CharConfidence method](#)

set_OMRValue

The `set_OMRValue` method sets the character values within the OMR field to indicate whether a check box is selected.

Description

Each character is assigned the value 0 (ASCII 48; not selected) or 1 (ASCII 49; selected) and the resulting binary string is expressed as a decimal. The following XML example represents a row of six check boxes and the binary string is 001000, which equals 8 in decimal. So, the required value is 8.

```
<F
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_dcdev580_1a">
  <V n="TYPE">1a</V>
  <V n="Position">120,1025,853,1169</V>
  <V n="STATUS">0</V>
  <V n="DensityString">BAAVAA</V>
  <C cn="10" cr="519,1132,559,1170">48</C>  <-- 48 indicates option not selected
  <C cn="10" cr="563,1132,603,1170">48</C>
  <C cn="10" cr="607,1132,647,1170">48</C>
  <C cn="10" cr="652,1132,692,1170">49</C>  <-- 49 indicates option selected
  <C cn="10" cr="697,1132,737,1170">48</C>
  <C cn="10" cr="763,1132,803,1170">48</C>
</F>
```

Syntax

C#

```
void set_OMRValue(int nIndex, int pVal)
```

Arguments

nIndex

0 specifies the primary value. Where the field contains multiple recognition, voting, or multi-pass data entry values, 1 specifies the first alternative, 2 specifies the second alternative, and so on.

pVal

A number that reflects the state of each OMR check box. (See the XML example.)

See also

[get_OMRValue](#), [get_CharValue](#)

Parent topic: [DCO methods](#)

set_Variable method

The `set_Variable` method sets the value of a referenced variable by using the name of the variable.

Description

.NET restriction: For VBScript, use the [Variable](#) instead.

If you set a variable that does not exist in the runtime object, this method automatically creates the variable.

Datacap uses a number of standard variables. For a listing of these variables, see the *Standard Variable Reference* in this IBM® Knowledge Center.

Syntax

C#

```
string set_Variable(string lpszName, string pVal)
```

Arguments

lpszName
The name of the variable, the value of which you want to set.

pVal
The value.

Applies to

All objects.

C# example

The following example assigns the value `IBM` to the variable `Company`.

```
m_oDCO.set_Variable("Company", "IBM")
```

The method either sets the value if the variable exists, or creates a new variable with the specified value. The following sample XML node shows the resulting variable and the value:

```
<V n="Company">IBM</V>
```

Parent topic: [DCO methods](#)

Related reference:

[AddVariable method](#)

[get_Variable method](#)

Related information:

[Standard Variable Reference](#)

Write method

The Write method saves the runtime batch object to a batch file, or a page object to a page file. When it writes a batch object, this method also writes all of the child objects, including documents, pages, and fields.

When you create a batch with an application that you developed in Datacap Studio, Datacap saves the batch as an XML file. The file is named after the last completed task in the workflow of the application. When you create a batch with an application that you developed outside of Datacap Studio, you can use the Write method to save the batch object to a file.

Syntax

VBScript

```
oDCO.Write(lpszFileName as String) as Boolean
```

C#

```
bool Write(string lpszFileName)
```

Applies to

Batch or page objects.

Arguments

lpszFileName
The full path and name of the output file.

Returns

Returns true if successful; returns false if unsuccessful.

C# example

This example writes the 20030086.003 batch object and the page objects of the batch to Verify.xml file.

```
bStatus = objBatch.Write ("c:\Datacap\MQSW\Batches\20030085.003\Verify.xml")
objPage.CreateFields
bStatus = objPage.Write ("c:\Datacap\BDOcs\Batches\20030085.003" & objPage.ID)
```

Parent topic: [DCO methods](#)

Related reference:

[Read method](#)

[WriteSetup method](#)

WriteSetup method

The WriteSetup method writes the Setup object and SetupNode objects to the Setup DCO file. You use this method after you read a setup DCO file from an external location.

Syntax

VBScript

```
oDCO.WriteSetup(FileName as String) as Boolean
```

C#

```
bool WriteSetup(string lpszFileName)
```

Applies to

Batch objects only.

Arguments

lpszFileName

Full path and name for the Setup DCO file (for example, C:\Datacap\APT\dco_APT\APT.xml).

Returns

Returns true if successful; Returns false if unsuccessful.

Example

See [ReadSetup](#) for an example.

Parent topic: [DCO methods](#)

Related reference:

[ReadSetup method](#)

[Write method](#)

DCOSetup API

The DCOSetup APIs include properties and methods that you can use to create or modify the document hierarchy. Datacap saves the document hierarchy as the setup DCO file in XML format, for example, C:\Datacap*application name*\dco_*application name*\dco_*application name*.xml.

In the setup DCO file, each document, page, and field type is defined as a separate node, as shown in this sample XML code:

```
<?xml-stylesheet type="text/xsl"
href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.develop.doc_...\dco.
xsl"?>
<S>
  <B type="batch_name"> 1

  [Variables and rules associated with this batch]

  </B>
  <D type="document_name"> 2

  [Variables and rules associated with this document]

  </D>

  [more document types]

  <P type="page_name"> 3

  [Variables and rules associated with this page]

  </P>

  [more page types]

  <F type="field_name"> 4

  [Variables and rules associated with this field]

  </F>

  [more field types]

  <DICTIONARY n="dict_name"> 5

  [Terms (words and values) in this dictionary]

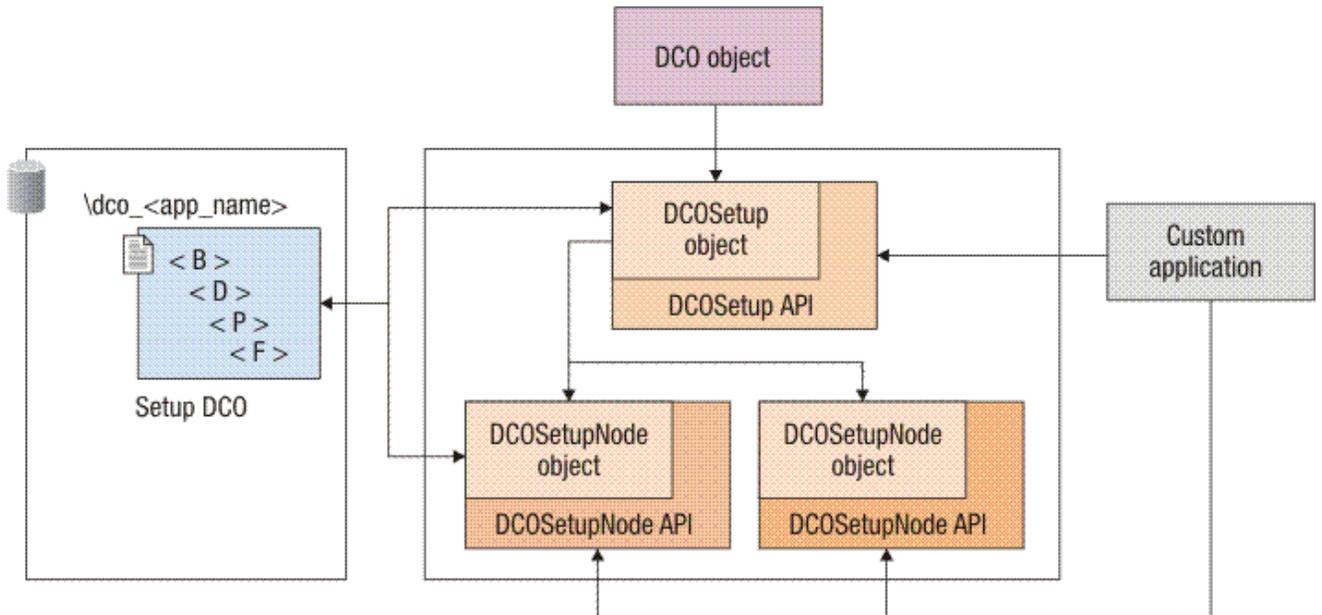
  </DICTIONARY>

  [more document types]
```

- 1 Setup node for the batch
- 2 Setup node for the document
- 3 Setup node for the page
- 4 Setup node for the field
- 5 Setup node for the dictionary

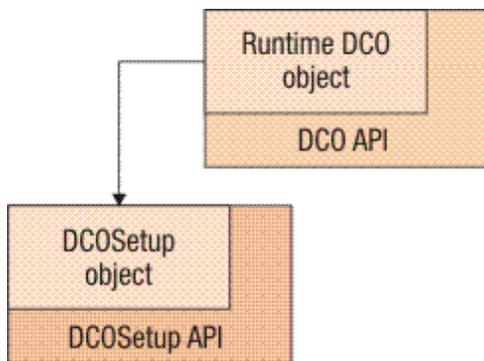
When you use an API to populate the DCOSetup object with child nodes from the Setup DCO file, each node becomes a separate DCOSetupNode object.

You use the DCOSetup APIs to modify the DCOSetup object and use the DCOSetupNode APIs to modify the child DCOSetupNode objects.



When you create a runtime DCO object (`m_oDCO` in this example), Datacap automatically creates a corresponding DCOSetup object and updates the setup DCO file.

```
TDCOLib.IDCO m_oDCO = new TDCOLib.DCOClass();
```



You can use the SetupObject method to access the DCOSetup object either directly or by first obtaining an interface to the DCOSetup object:

Accessing the DCOSetup Object Directly

```
m_oDCO.SetupObject().AddNode(1, "NewDoc1");
```

Accessing the DCOSetup object through an interface

```
TDCOLib.DCOSetup m_oDCOSetup = m_oDCO.SetupObject();
//Obtaining an interface

m_oDCOSetup.AddNode(1, "NewDoc2");
//Accessing the DCOSetup object through the interface
```

The DCOSetup object is initially null. You populate the DCOSetup object and the child objects in one of the following ways:

Use the ReadSetup method from an existing Setup DCO file

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
```

The ReadSetup method parses the XML file and creates all of the child DCOSetupNode objects, such as documents, pages, and fields, that are defined for the document hierarchy.

Use the AddNode method

```
m_oDCO.SetupObject().AddNode(1, "NewDoc1");
```

If you are creating a Setup DCO or modifying an existing Setup DCO, you can write the memory resident object hierarchy to disk by using the WriteSetup method:

```
m_oDCO.WriteSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
```

- [DCOSetup properties](#)
You can use DCOSetup properties to access DCOSetup objects and modify metadata, including dictionary paths, terms, names, and key values that are used in verification tasks.
- [DCOSetup methods](#)
DCOSetup methods enable access to DCO nodes and dictionaries. You can use DCOSetup methods to obtain and modify metadata, including names, values, terms, and quantities. You can also use these methods to create or delete nodes.

Parent topic: [Datacap object API reference](#)

Related reference:

[AddNode method](#)

[ReadSetup method](#)

[SetupObject method](#)

[WriteSetup method](#)

DCOSetup properties

You can use DCOSetup properties to access DCOSetup objects and modify metadata, including dictionary paths, terms, names, and key values that are used in verification tasks.

- [DictionaryName property](#)
The DictionaryName property sets or gets the name of a dictionary in the Setup DCO.
- [Path property](#)
The Path property sets or gets the path to the setup DCO file.
- [Value property](#)
The Value property sets or gets the key value of a dictionary term in the Setup DCO file. You can use this property when you need to pass a value to a variable.
- [Word property](#)
The Word property sets or gets the word value in a dictionary term in the setup DCO file. You can use this property to pass a value to a variable.

Parent topic: [DCOSetup API](#)

DictionaryName property

The DictionaryName property sets or gets the name of a dictionary in the Setup DCO.

VBScript only: Because extended properties are not supported through the C# .NET Interop interface, you must instead use the [set_DictionaryName method](#) or the [get_DictionaryName method](#).

Syntax

VBScript

```
oSO.DictionaryName(DictIndex as Long) as String
```

Arguments

DictIndex

The number that indicates the location of the dictionary in the setup DCO, with 0 being the first dictionary, 1 being the second, and so on.

Type

Read and write.

VBScript example

This example sets the name of the first dictionary in the setup DCO file to `Vendors` and the name of the second dictionary to `Customers`.

```
oSO.DictionaryName(0) = "Vendors"  
oSO.DictionaryName(1) = "Customers"
```

Parent topic: [DCOSetup properties](#)

Related reference:

[Value property](#)

[Word property](#)

Path property

The Path property sets or gets the path to the setup DCO file.

Syntax

VBScript

```
oSO.Path as String
```

C#

```
string Path { set; get; }
```

Type

Read and write.

VBScript example

This example sets the path to the BDCcs.xml setup DCO file:

```
oSO.Path = "c:\Datacap\BDOcs\dco_BDOcs\BDOcs.xml"
```

Parent topic: [DCOSetup properties](#)

Related reference:

Value property

The Value property sets or gets the key value of a dictionary term in the Setup DCO file. You can use this property when you need to pass a value to a variable.

VBScript restriction: Because extended properties are not supported through the C# .NET Interop interface, you must instead use the [set_Value method](#) or the [get_Value method](#).

The following example defines one dictionary with four words, where `RoutingInstructions` is the dictionary name. The text in each set of quotation marks is the key value that is associated with the word that follows it. Datacap uses the key value to identify words in the dictionary that are displayed to an operator during a verification task. Datacap populates the associated field with the word that the operator selects.

```
<DICTIONARY n="RoutingInstructions">
  <WORD v="None">None</WORD>
  <WORD v="Delete">Delete</WORD>
  <WORD v="Rescan">Rescan</WORD>
  <WORD v="Review">Review</WORD>
</DICTIONARY>
```

Syntax

VBScript

```
oSO.Value(nDictionary subscript as Long, nIndex subscript as Long) as String
```

Type

Read and write.

Arguments

`nDictionary`

The index of the dictionary in the setup object, where 0 is the first dictionary.

`nIndex`

The index of the entry in the dictionary, where 0 is the first entry.

VBScript example

This example sets the key value of the seventh entry in the first dictionary to 123.

```
oSO.Value(0,6) = "123"
```

Parent topic: [DCOSetup properties](#)

Related reference:

[DictionaryName property](#)

[Word property](#)

Word property

The Word property sets or gets the word value in a dictionary term in the setup DCO file. You can use this property to pass a value to a variable.

VBScript restriction: Because extended properties are not supported through the C# .NET Interop interface, you must instead use [set_Word](#) or [get_Word](#).

The following sample setup DCO XML file defines one dictionary with four words, where `RoutingInstructions` is the dictionary name. The text in each set of quotation marks is the key value that is associated with the word that follows it.

```
<DICTIONARY n="RoutingInstructions">
  <WORD v="None">None</WORD>
  <WORD v="Delete">Delete</WORD>
  <WORD v="Rescan">Rescan</WORD>
  <WORD v="Review">Review</WORD>
</DICTIONARY>
```

Datacap uses the key value to identify words in the dictionary that are displayed to an operator during a verification task. Datacap populates the associated field with the word that the operator selects.

Syntax

VBScript

```
oSO.Word(nDictionary as Long, nIndex as Long) as String
```

Type

Read and write.

Arguments

`nDictionary`

The index of the dictionary within the Setup object, where 0 is the first dictionary.

`nIndex`

The index of the entry within the dictionary, where 0 is the first entry.

VBScript example

This example compares the value of the `VendorName` field to the seventh entry in the first dictionary in the Setup DCO file:

```
oSO.Word(0,6) = "FreightLiners"
If VendorName = oSO.Word(0,6) then...
```

Parent topic: [DCOSetup properties](#)

Related reference:

[DictionaryName property](#)

[Value property](#)

DCOSetup methods

DCOSetup methods enable access to DCO nodes and dictionaries. You can use DCOSetup methods to obtain and modify metadata, including names, values, terms, and quantities. You can also use these methods to create or delete nodes.

- [AddNode method](#)

The `AddNode` method adds a node to the Setup DCO object at the end of the section for the type specified (batch, document, page, field, character).

- [DeleteNode method](#)
The `DeleteNode` method reads the value of the `nIndex` argument and deletes the corresponding node from the Setup DCO.
- [DeleteNodeByName Method](#)
The `DeleteNodeByName` method deletes a node from the Setup DCO by referencing the node's name.
- [get_DictionaryName method](#)
The `get_DictionaryName` method gets the name of a dictionary from the Setup DCO.
- [get_Value method](#)
The `get_Value` method gets the key value of a term in a dictionary from the Setup DCO.
- [get_Word method](#)
The `get_Word` method gets the word value from a dictionary term.
- [GetNode method](#)
The `GetNode` method gets an interface to a `SetupNode` object in the Setup DCO by using an index value. You can use this method when you know the position of the object but not the name.
- [GetNodeByName method](#)
The `GetNodeByName` method accesses a `SetupNode` object from the Setup DCO by using the object name.
- [NumOfDictionaries method](#)
The `NumOfDictionaries` method returns the number of dictionaries that are defined in the Setup DCO.
- [NumOfNodes method](#)
The `NumOfNodes` method gets the number of documents, pages, fields, or character nodes in the Setup DCO.
- [NumOfWords](#)
The `NumOfWords` method returns the number of words that are defined in the dictionary in the Setup DCO.
- [ReadLock](#)
The `ReadLock` method locks, and prevents other applications from writing to, the specified Setup DCO file.
- [ReadSetup](#)
This method is identical to `DCO.ReadSetup`, except that you call it from a `DCOSetup` object.
- [set_DictionaryName method](#)
The `set_DictionaryName` method sets the name of a dictionary within the Setup DCO.
- [set_Value method](#)
The `set_Value` method sets the key value in a dictionary term.
- [set_Word method](#)
The `set_Word` method sets the word value in a dictionary term.
- [ShowSetupDialog](#)
The `ShowSetupDialog` method is identical to `DCO.ShowSetupDialog`, except that you call it from a `DCOSetup` object.
- [UnlockIt](#)
The `UnlockIt` method unlocks the Setup DCO file that was locked previously with the `ReadLock` method.
- [WriteSetup method](#)
The `WriteSetup` method is identical to `DCO.WriteSetup`, except that you call it from a `DCOSetup` object.

Parent topic: [DCOSetup API](#)

AddNode method

The `AddNode` method adds a node to the Setup DCO object at the end of the section for the type specified (batch, document, page, field, character).

Syntax

VBScript

```
oSO.AddNode (Type as Long, Name as String) as Boolean
```

C#

```
bool AddNode(int nType, string lpszNodeName)
```

Arguments

nType

Specify one of the following number values to indicate the type of node to add:

0 = Batch

1 = Document

2 = Page

3 = Field

4 = Character

lpszNodeName

The name of the new node. This name is referenced as the node's `type` attribute in the Setup DCO file, as shown in this example:

```
<D type="NewDocNode">
```

Returns

Returns true if successful; returns false if unsuccessful.

C# example

The following example adds a document node to the Setup DCO:

```
m_oDCOSetup.AddNode(1, "NewDocNode");
```

The new node inherits its attributes and variables from the object template in the file `C:\Datacap\dcshared\dcotemp.xml`. The resulting Setup object is shown in this sample:

```
<D type="NewDocNode">  
  <V n="ID">0</V>  
  <V n="TYPE">Document</V>  
  <V n="STATUS">0</V>  
  <V n="DOC DATA">0</V>  
  <V n="MIN_TYPES">0</V>  
  <V n="MAX_TYPES">0</V>  
</D>
```

Parent topic: [DCOSetup methods](#)

Related reference:

[DeleteNode method](#)

[WriteSetup method](#)

DeleteNode method

The `DeleteNode` method reads the value of the `nIndex` argument and deletes the corresponding node from the Setup DCO.

Syntax

VBScript

```
oSO.DeleteNode(nType as Long, nIndex subscript as Long) as Boolean
```

C#

```
bool DeleteNode(int nType, int nIndex)
```

Arguments

nType

The type of the node to delete:

0 = Batch

1 = Document

2 = Page

3 = Field

4 = Character

nIndex

The index of the node to delete, where 0 is the first node of the specified type.

Returns

Returns true if successful; returns false if unsuccessful.

C# example

The following example deletes the third document node (type = 1, index = 2) from the Setup DCO:

```
m_oDCOSetup.DeleteNode(1, 2);
```

Parent topic: [DCOSetup methods](#)

Related reference:

[AddNode method](#)

[DeleteNodeByName Method](#)

DeleteNodeByName Method

The `DeleteNodeByName` method deletes a node from the Setup DCO by referencing the node's name.

Description

`DeleteNodeByName` references a node through the node's name, and deletes the node.

Syntax

VBScript

```
oSO.DeleteNode(nType as Long, lpszName as String) as Boolean
```

C#

```
bool DeleteNodeByName(int nType, string lpszName)
```

Arguments

nType

The type of the node to delete:

0 = Batch

1 = Document

2 = Page

3 = Field

4 = Character

lpszName

The name of the node to delete, as referenced in the node's type attribute:

```
<D type="NewDocNode">
```

Returns

True if successful; False otherwise.

VBScript Example

```
Dim OK
OK = oSO.DeleteNodeByName(3, "Total")
msgbox OK
msgbox "The number of fields remaining is " & oSO.NumOfNodes(3)
```

C# Example

The following example deletes the document node, "NewDocNode," from the Setup DCO:

```
m_oDCOSetup.DeleteNode(1, "NewDocNode");
```

See Also

[DeleteNode method](#), [AddNode method](#)

Parent topic: [DCOSetup methods](#)

get_DictionaryName method

The get_DictionaryName method gets the name of a dictionary from the Setup DCO.

.NET restriction: For VBScript, use the [DictionaryName property](#) instead.

Syntax

C#

```
string get_DictionaryName(int nDictionary)
```

Arguments

nDictionary

The index of the dictionary within the Setup DCO, where 0 is the first dictionary.

Returns

The name of the dictionary.

C# example

This example gets the name of the first dictionary from the Setup DCO file, APT.XML:

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");  
string strDictName = m_oDCO.SetupObject().get_DictionaryName(0);
```

Parent topic: [DCOSetup methods](#)

Related reference:

[ReadSetup method](#)

[SetupObject method](#)

[set_DictionaryName method](#)

[WriteSetup method](#)

get_Value method

The `get_Value` method gets the key value of a term in a dictionary from the Setup DCO.

.NET restriction: For VBScript, use the [Value property](#) instead.

In this sample XML code, `RoutingInstructions` is the name of the dictionary, which is followed by the key value and word pairs ("None",None, and so on).

```
<DICT n="RoutingInstructions">  
  <W v="None">None</W>  
  <W v="Delete">Delete</W>  
  <W v="Rescan">Rescan</W>  
  <W v="Review">Review</W>  
</Dict>
```

Syntax

C#

```
string get_Value(int nDictionary, int nIndex)
```

Arguments

`nDictionary`

The index of the dictionary within the Setup object, where 0 is the first dictionary.

`nIndex`

The index of the term within the dictionary, where 0 is the first term.

C# example

This example gets the key value on the second term in the first dictionary:

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");  
string strDictValue = m_oDCO.SetupObject().get_Value(0, 1);
```

Parent topic: [DCOSetup methods](#)

Related reference:

[set_Value method](#)

get_Word method

The `get_Word` method gets the word value from a dictionary term.

.NET restriction: For VBScript, use the [Word property](#) instead.

In the following XML sample, `RoutingInstructions` is the name of the dictionary, which is followed by the key value and word value pairs ("None", None, and so on):

```
<DICT n="RoutingInstructions">
  <W v="None">None</W>
  <W v="Delete">Delete</W>
  <W v="Rescan">Rescan</W>
  <W v="Review">Review</W>
</Dict>
```

Syntax

C#

```
string get_Word(int nDictionary, int nIndex)
```

Arguments

`nDictionary`

The index of the dictionary within the Setup object, where 0 is the first dictionary

`nIndex`

The index of the term within the dictionary, where 0 is the first term

C# example

This example gets the word value on the second term in the first dictionary:

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
string strDictWord = m_oDCO.SetupObject().get_Word(0, 1);
```

Parent topic: [DCOSetup methods](#)

Related reference:

[set_Word method](#)

GetNode method

The `GetNode` method gets an interface to a `SetupNode` object in the Setup DCO by using an index value. You can use this method when you know the position of the object but not the name.

Syntax

VBScript

```
oSO.GetNode(nType as Long, nIndex as Long) child node as node object
```

C#

```
TDCOLib.DCOSetupNode GetNode(int nType, int nIndex)
```

Arguments

nType

Value indicating the component type:

0 = Batch

1 = Document

2 = Page

3 = Field

nIndex

Value of the node's index, where 0 is the first node of the specified type in the Setup DCO

Returns

The SetupNode object on success; nothing (null) on failure.

C# example

The following example gets an interface to the first node of type `page` from the Setup DCO:

```
m_oDCOSetupNode = m_oDCOSetup.GetNode(2, 0);
```

Parent topic: [DCOSetup methods](#)

Related reference:

[GetNodeByName method](#)

GetNodeByName method

The `GetNodeByName` method accesses a `SetupNode` object from the Setup DCO by using the object name.

Syntax

VBScript

```
oSO.GetNodeByName(nType as Long, lpszName as String) as child node as node object
```

C#

```
TDCOLib.DCOSetupNode GetNodeByName(int nType, string lpszName)
```

Arguments

nType

Value indicating the component type:

0 = Batch

1 = Document

2 = Page

3 = Field

lpszName

The node's name, as referenced in the node's `type` attribute:

```
<P type="Main_Page">
```

Returns

The SetupNode object on success; nothing (null) on failure.

C# Example

The following example gets an interface to the node "Main_Page" of type "page" from the Setup DCO:

```
m_oDCOSetupNode = m_oDCOSetup.GetNodeByName(2, "Main_Page");
```

See also

[GetNode method](#)

Parent topic: [DCOSetup methods](#)

NumOfDictionaries method

The NumOfDictionaries method returns the number of dictionaries that are defined in the Setup DCO.

```
<DICTIONARIES n="RoutingInstructions">
  <W v="None">None</W>
  <W v="Delete">Delete</W>
  <W v="Rescan">Rescan</W>
  <W v="Review">Review</W>
</DICTIONARIES>
```

In the sample XML code, there is one dictionary that is called RoutingInstructions.

Syntax

VBScript

```
oSO.NumOfDictionaries( ) as Long
```

C#

```
int NumOfDictionaries()
```

Arguments

None

Returns

The number of dictionaries defined in this Setup DCO.

See also

[NumOfWords](#), [DictionaryName](#) property

Parent topic: [DCOSetup methods](#)

NumOfNodes method

The `NumOfNodes` method gets the number of documents, pages, fields, or character nodes in the Setup DCO.

Syntax

VBScript

```
oSO.NumOfNodes(nType as Long) as Long
```

C#

```
int NumOfNodes(int nType)
```

Arguments

`nType`

Value indicating the object type:

0 = Batch

1 = Document

2 = Page

3 = Field

4 = Character

Returns

Value specifying the number of nodes of the specified type.

VBScript Example

This example reads the Setup DCO and gets the number of field nodes:

```
Dim FlDs
Call oSO.ReadSetup("C:\Datacap\APT\dco_APT\APT.xml")
nFlDs = oSO.NumOfNodes(3)
msgbox nFlDs
```

C#

This example reads the Setup DCO and gets the number of field nodes:

```
m_oDCO.ReadSetup("C:\Datacap\APT\dco_APT\APT.xml");
TDCOLib.DCOSetup m_oDCOSetup = m_oDCO.SetupObject();
int numNodes = m_oDCOSetup.NumOfNodes(3);
```

See also

[AddNode method](#), [get_DictionaryName method](#), [GetNodeByName method](#), [DeleteNode method](#)

Parent topic: [DCOSetup methods](#)

NumOfWords

The `NumOfWords` method returns the number of words that are defined in the dictionary in the Setup DCO.

Description

The following example defines one dictionary with four words, where `RoutingInstructions` is the dictionary name, and the text in each set of quotation marks is the key value that is associated with the word that follows it.

```
<DICTIONARY n="RoutingInstructions">
  <WORD v="None">None</WORD>
  <WORD v="Delete">Delete</WORD>
  <WORD v="Rescan">Rescan</WORD>
  <WORD v="Review">Review</WORD>
</DICTIONARY>
```

Syntax

VBScript

```
oSO.NumOfWords (nDictionary subscript as Long) as Long
```

C#

```
int NumOfWords(int nDictionary)
```

Arguments

`nDictionary`

The index of the dictionary, where 0 is the first dictionary in the Setup DCO.

Returns

The number of words in the dictionary.

See also

[NumOfDictionaries](#)

Parent topic: [DCOSetup methods](#)

ReadLock

The `ReadLock` method locks, and prevents other applications from writing to, the specified Setup DCO file.

Description

`ReadLock` attempts to obtain the lock every 50 milliseconds. After 300 unsuccessful attempts, `ReadLock` stops the attempt and returns `False`. The attempt fails when another process locked the file.

Syntax

C#

```
bool ReadLock(string lpszPath)
```

Arguments

`lpszPath`

Full path and file name to the Setup DCO file.

Returns

True if successful; False otherwise.

See also

[UnlockIt](#)

Parent topic: [DCOSetup methods](#)

ReadSetup

This method is identical to `DCO.ReadSetup`, except that you call it from a `DCOSetup` object.

See [DCO.ReadSetup](#) for details.

Parent topic: [DCOSetup methods](#)

set_DictionaryName method

The `set_DictionaryName` method sets the name of a dictionary within the Setup DCO.

Syntax

C#

```
void set_DictionaryName(int nDictionary, string pVal)
```

Arguments

`nDictionary`

The index of the dictionary within the Setup DCO, where 0 is the first dictionary.

`pVal`

The new dictionary name.

Returns

Nothing.

C# example

This C# example sets the name of the first dictionary in the Setup DCO file `APT.XML`:

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");  
string strDictName = m_oDCO.SetupObject().set_DictionaryName(0, "Datacap");  
m_oDCO.WriteSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
```

Parent topic: [DCOSetup methods](#)

Related reference:

[get_DictionaryName method](#)

[ReadSetup method](#)

[SetupObject method](#)

[WriteSetup method](#)

set_Value method

The set_Value method sets the key value in a dictionary term.

.NET restriction: For VBScript, use the [Value](#) instead.

The following example defines one dictionary with four words, where `RoutingInstructions` is the dictionary name. The text in each set of quotation marks is the key value that is associated with the word that follows it.

```
<DICTIONARY n="RoutingInstructions">
  <WORD v="None">None</WORD>
  <WORD v="Delete">Delete</WORD>
  <WORD v="Rescan">Rescan</WORD>
  <WORD v="Review">Review</WORD>
</DICTIONARY>
```

Syntax

C#

```
void set_Value(int nDictionary, int nIndex, string pVal)
```

Arguments

nDictionary

The index of the dictionary within the Setup object, where 0 is the first dictionary.

nIndex

The index of the term within the dictionary, where 0 is the first term.

pVal

The key value.

Returns

Nothing

C# example

This example sets the key value on the second term in the first dictionary to `Delete`:

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
m_oDCO.SetupObject().set_Value(0, 1, "Delete");
m_oDCO.WriteSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
```

Parent topic: [DCOSetup methods](#)

Related reference:

[get_Value method](#)

set_Word method

The set_Word method sets the word value in a dictionary term.

.NET restriction: For VBScript, use the [Word](#) instead.

The following example defines one dictionary with four words, where `RoutingInstructions` is the dictionary name. The text in each set of quotation marks is the key value that is associated with the word that follows it.

```
<DICT n="RoutingInstructions">
  <W v="None">None</W>
  <W v="Delete">Delete</W>
  <W v="Rescan">Rescan</W>
  <W v="Review">Review</W>
</DICT>
```

Syntax

C#

```
void set_Word(int nDictionary, int nIndex, string pVal)
```

Arguments

nDictionary

The index of the dictionary within the Setup object, where 0 is the first dictionary.

nIndex

The index of the term within the dictionary, where 0 is the first term.

pVal

The key value.

C# example

This example sets the word value on the second term in the first dictionary to `Delete`:

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
m_oDCO.SetupObject().set_Word(0, 1, "Delete");
m_oDCO.WriteSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
```

Parent topic: [DCOSetup methods](#)

Related reference:

[get_Word method](#)

ShowSetupDialog

The `ShowSetupDialog` method is identical to `DCO.ShowSetupDialog`, except that you call it from a `DCOSetup` object.

Parent topic: [DCOSetup methods](#)

UnlockIt

The `UnlockIt` method unlocks the Setup DCO file that was locked previously with the `ReadLock` method.

Syntax

C#

```
bool UnlockIt()
```

Arguments

None

Returns

True if successful; False otherwise.

See also

[ReadLock](#)

Parent topic: [DCOSetup methods](#)

WriteSetup method

The `WriteSetup` method is identical to `DCO.WriteSetup`, except that you call it from a `DCOSetup` object.

For more information, see [DCO.WriteSetup](#).

Parent topic: [DCOSetup methods](#)

DCOSetupNode APIs

You can use the `DCOSetupNode` API properties and methods to access and modify Setup DCO child objects, including rules and variables.

Each runtime DCO object inherits attributes and variables from the corresponding `DCOSetupNode` object. For example, as demonstrated in the following setup node sample, all documents that are identified as `Invoice` documents include the `TYPE`, `STATUS`, `DOC DATA`, `MIN_TYPES`, `MAX_TYPES`, and `rules` variables. The `Invoice` document object must also contain at least one `Main_Page` page object. Similarly, all runtime objects that are identified as `Main_Page` types must be in the first position of the `Invoice` document.

```
<D type="Invoice"> 1
  <V n="ID">0</V>
  <V n="TYPE">Document</V>
  <V n="STATUS">0</V>
  <V n="DOC DATA">0</V>
  <V n="MIN_TYPES">0</V> 2
  <V n="MAX_TYPES">0</V>
  <V n="rules"><in><r id="1" rs="11" />
  /><r id="5" rs="7" /><r
  id="4" rs="16" /></in></V>

  <P type="Main_Page" pos="1" min="1" max="1"/>
  <P type="Trailing_Page" pos="0" min="0" max="0"/> 3
  <P type="Attachment" pos="0" min="0" max="0"/>
</D>
```

1 Setup node for the document type `Invoice`

2 Variables

3 Parameters

`DCOSetupNode` objects are created and accessed through the parent `DCOSetup` object.

The `SetupNode` object also defines the document integrity rules for that object type. The following example specifies that a Tax Return document must have at least one (`min="1"`) and at most one (`max="1"`) page of type `1040EZ`. The relative position attribute (`pos="2"`) specifies that the `1040EZ` page must come after the `TaxSep` page (if present) and before the `Attachment` page (if present):

```

<D type="Tax Return">
  <V n="ID">0</V>
  <V n="ID">0</V>
  <V n="TYPE">Document</V>
  <V n="STATUS">0</V>
  <V n="DOC_DATA">0</V>
  <V n="MIN_TYPES">2</V>
  <V n="MAX_TYPES">0</V>
  <V n="rules"></V>
  <P type="TaxSep" pos="1" min="0" max="1"/>
  <P type="1040EZ" pos="2" min="1" max="1"/>      <--- Rule for page 1040EZ
  <P type="Attachment" pos="3" min="0" max="0"/>
</D>

```

The `max="0"` value means unlimited. In the example, there can be any number of attachment pages. The `MIN_TYPES` and `MAX_TYPES` variables define the minimum and maximum number of child object types that must be present at run time to constitute a valid document. The `MIN_TYPES` value (2) specifies that each runtime document must have at least two different page types.

After you obtain an interface to the `DCOSetup` object, you can obtain an interface to any `DCOSetupNode` object by using the [GetNode method](#) or the [GetNodeByName method](#), as shown in the following example:

```
TDCOLib.DCOSetupNode m_oDCOSetupNode = m_oDCOSetup.GetNodeByName(2, "Main_Page");
```

Through the `DCOSetup` object interface, you can access all of the `DCOSetupNode` object properties and methods.

- [DCOSetupNode properties](#)
You can use `DCOSetupNode` properties to access `DCOSetup` objects and modify metadata, including names, types, and values. You can also use these properties to modify rules, variables, and child objects.
- [DCOSetupNode methods](#)
You can use `DCOSetupNode` methods to add or delete rules and variables. You can also use these methods to obtain and modify metadata, including names, number attributes, object types, and values. You might use these methods when the business requirements of your organization change and require different rules and variables.

Parent topic: [Datacap object API reference](#)

DCOSetupNode properties

You can use `DCOSetupNode` properties to access `DCOSetup` objects and modify metadata, including names, types, and values. You can also use these properties to modify rules, variables, and child objects.

- [Name property](#)
The `Name` property sets or gets the name of the `SetupNode` object.
- [ObjectType property](#)
The `ObjectType` property sets or gets the `SetupNode` object type, such as a batch, document, page, field, or character.
- [RuleChildName](#)
The `RuleChildName` property sets or gets the name of a rule that is referenced by an index.
- [RuleMaxNum](#)
The `RuleMaxNum` property sets or gets the `max` attribute for a rule.
- [RuleMinNum](#)
The `RuleMinNum` property sets or gets the `min` attribute for a rule.
- [RuleObjectType](#)
The `RuleObjectType` property sets or gets the type of an object.

- [RulePosition](#)
The `RulePosition` property determines the order of rules, and optionally restricts the order of child objects within a parent object at run time.
- [Variable](#)
The `Variable` property sets or gets the value that is associated with a variable in the `SetupNode` object. You can use this property to identify objects (for example, pages) that require special processing when a variable contains a specific value.
- [VariableName](#)
The `VariableName` property sets or gets the name of a variable that is accessed by an index in the `DCOSetupNode` object. You can use this property to identify objects (for example, pages) that require special processing when a variable contains a specific value. This property is useful when a node contains many child objects and variables.
- [VariableValue](#)
The `VariableValue` property sets or gets the value of an indexed variable from the `DCOSetupNode` object. You can use this property to identify objects (for example, pages) when a node contains many child objects and variables.

Parent topic: [DCOSetupNode APIs](#)

Name property

The `Name` property sets or gets the name of the `SetupNode` object.

The `SetupNode` object name is saved as the `type` attribute in the Setup DCO XML file:

```
<D type="NodeName">
```

The name is the identifier that is used to refer to child objects within the Setup DCO, as shown in this example:

```
<P type="Main_Page">
.
.
.
  <F type="Vendor" pos="0" min="0" max="0"/>  <!--Referencing the Vendor node by
name
```

Syntax

VBScript

```
oSNO.Name as String
```

C#

```
string Name { set; get; }
```

Type

Read and write.

VBScript example

In the following example, the code populates a setup object from the `BDOcs.xml` setup DCO file. The code then uses the `Name` property to update the name of the second node of type `Page` to `Invoice` in the `Setup` object. The example also writes the updated Setup DCO file to disk.

```

Call oSO.ReadSetup("c:\Datacap\BDOcs\dco_BDOcs\BDOcs.xml")
Dim objInvPage
Set objInvPage = oSO.GetNode(2,1)
objInvPage.Name = "Invoice"
Call oSO.WriteSetup("c:\Datacap\BDOcs\dco_BDOcs\BDOcs.xml")

```

C# example

In the following example, after the code populates a setup object from the APT.XML setup DCO file, the code uses the `Name` property to update the name of the `TM000001` node to `Invoice` in the Setup object. The example also writes the updated Setup DCO file to disk.

```

m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
TDCOLib.DCOSetupNode m_oDCOSetupNode = m_oDCO.FindChild("TM000001").SetupNode();
m_oDCOSetupNode.Name = "Invoice";
m_oDCO.WriteSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");

```

Parent topic: [DCOSetupNode properties](#)

Related reference:

[AddNode method](#)

[GetNode method](#)

[GetNodeByName method](#)

ObjectType property

The `ObjectType` property sets or gets the `SetupNode` object type, such as a batch, document, page, field, or character.

Use one of these number values to determine the `SetupNode` object type:

Object type

- 0 = Batch.
- 1 = Document.
- 2 = Page.
- 3 = Field.
- 4 = Character.

Syntax

VBScript

```
oSNO.ObjectType as Long
```

C#

```
int ObjectType { set; get; }
```

Type

Read and write.

VBScript example

This example gets a `SetupNode` object for the `Total` field and sends the type (3 = Field) to a script debugger:

```
Dim objTotal
Set objTotal = oSO.GetNodeByName(3, "Total")
Debug.write objTotal.ObjectType
```

Parent topic: [DCOSetupNode properties](#)

Related reference:

[GetNodeByName method](#)

[RuleObjectType](#)

RuleChildName

The RuleChildName property sets or gets the name of a rule that is referenced by an index.

Description

Important: VBScript only. Extended properties are not supported through the C# .NET Interop interface.

Instead, use [set_RuleChildName](#) or [get_RuleChildName](#).

For more information, see [get_RuleChildName](#).

Syntax

VBScript

```
oSNO.RuleChildName(nIndex as Long) as String
```

Arguments

nIndex

The index of the rule, where 0 is the first rule.

Type

Read and write

Example

VBScript

```
Dim objDoc
Set objDoc = MQDCO.GetNodebyName(1, "Doc")
Call objDoc.AddRule (2, "Page1", 1, 1, 3)
msgbox "The name of Rule 1 is " & objDoc.RuleChildName(0)
```

See also

[AddRule](#)

Parent topic: [DCOSetupNode properties](#)

RuleMaxNum

The RuleMaxNum property sets or gets the max attribute for a rule.

Description

Important: VBScript only. Extended properties are not supported through the C# .NET Interop interface. Instead, use [set_RuleMaxNumber](#) or [get_RuleMaxNumber](#). For more information, see [get_RuleMaxNumber](#).

Syntax

VBScript

```
oSNO.RuleMaxNum (RuleIndex as Long)
```

Arguments

RuleIndex

The index of the rule, where 0 is the first rule under this SetupNode object.

Type

Read and write

Example

VBScript

```
Dim MaxTotal  
MaxTotal = objInv.RuleMaxNum(3)  
msgbox MaxTotal
```

See also

[RuleMinNum](#)

Parent topic: [DCOSetupNode properties](#)

RuleMinNum

The RuleMinNum property sets or gets the `min` attribute for a rule.

Description

Important: VBScript only. Extended properties are not supported through the C# .NET Interop interface. Instead, use [set_RuleMinNumber](#) or [get_RuleMinNumber](#).

For more information, see [get_RuleMinNumber](#).

Syntax

VBScript

```
oSNO.RuleMinNum (RuleIndex as Long) as Long
```

Arguments

RuleIndex

The index of the rule, where 0 is the first rule under this SetupNode object.

Type

Read and write

Example

VBScript

```
Dim MinSSN
MinSSN = objPageTwo.RuleMinNum(0)
msgbox MinSSN
```

See also

[RuleMaxNum](#)

Parent topic: [DCOSetupNode properties](#)

RuleObjectType

The RuleObjectType property sets or gets the type of an object.

Description

Important: VBScript only. Extended properties are not supported through the C# .NET Interop interface. Instead, use [set_RuleObjectType](#) or [get_RuleObjectType](#).

The object type is determined by a value 0-3.

0 = Batch

1 = Document

2 = Page

3 = Field

Syntax

VBScript

```
oSNO.RuleObjectType (Index as Long) as Long
```

Arguments

Index

The index of the rule, where 0 is the first rule under this SetupNode object.

Type

Read and write

Example

VBScript

```
Dim TypeSSN
TypeSSN = objPageTwo.RuleObjType(0)
msgbox TypeSSN
```

See also

[get_RuleObjectType](#)

Parent topic: [DCOSetupNode properties](#)

RulePosition

The RulePosition property determines the order of rules, and optionally restricts the order of child objects within a parent object at run time.

Description

Important: VBScript only. Extended properties are not supported through the C# .NET Interop interface. Instead, use [set_RulePosition](#) or [get_RulePosition](#).

All child objects with rules that specify `Position=1` must appear before any child objects with higher position values (greater than 1). A `Position=0` means that the child item can appear in any order.

Syntax

VBScript

```
oSNO.RulePosition (Index as Long) as Long
```

Type

Read and write

Arguments

Index

The index of the rule, where 0 is the first rule under this SetupNode object.

Parent topic: [DCOSetupNode properties](#)

Variable

The Variable property sets or gets the value that is associated with a variable in the SetupNode object. You can use this property to identify objects (for example, pages) that require special processing when a variable contains a specific value.

Description

Important: VBScript only. Extended properties are not supported through the C# .NET Interop interface. Instead, use [set_Variable](#) or [get_Variable](#).

Syntax

VBScript

```
oSNO.Variable (VarName as String) as Variant
```

Type

Read and write

Arguments

VarName

Name of the variable (case sensitive)

Example

VBScript

```
Call objTotal.AddVariable ("Length", "10")  
msgbox objTotal.Variable("Length")
```

See also

[get_ Variable](#)

Parent topic: [DCOSetupNode properties](#)

VariableName

The `VariableName` property sets or gets the name of a variable that is accessed by an index in the `DCOSetupNode` object. You can use this property to identify objects (for example, pages) that require special processing when a variable contains a specific value. This property is useful when a node contains many child objects and variables.

Description

Important: VBScript only. Extended properties are not supported through the C# .NET Interop interface. Instead, use [set_ VariableName](#) or [get_ VariableName](#).

Syntax

VBScript

```
oSNO.VariableName (Node as Long) as String
```

Type

Read and write.

Arguments

Node

The index of the variable, where 0 is the first variable that is defined in the `DCOSetupNode` object.

See also

[get_ VariableName](#)

Parent topic: [DCOSetupNode properties](#)

VariableValue

The `VariableValue` property sets or gets the value of an indexed variable from the `DCOSetupNode` object. You can use this property to identify objects (for example, pages) when a node contains many child objects and variables.

Description

Important: VBScript only. Extended properties are not supported through the C# .NET Interop interface. Instead, use [set_RuleObjectType](#) or [get_RuleObjectType](#).

Syntax

VBScript

```
oSNO.VariableValue (Node as Long) as Variant
```

Arguments

Node

The index of the variable, where 0 is the first variable that is defined in the `DCOSetupNode` object.

Data type

Variant

Example

VBScript

This example displays the current value that is assigned to the third variable that is linked to the `objTotal` node:

```
msgbox objTotal.VariableValue(2)
```

read/write

Read and write

See also

[get_RuleObjectType](#)

Parent topic: [DCOSetupNode properties](#)

DCOSetupNode methods

You can use `DCOSetupNode` methods to add or delete rules and variables. You can also use these methods to obtain and modify metadata, including names, number attributes, object types, and values. You might use these methods when the business requirements of your organization change and require different rules and variables.

- [AddRule method](#)
The `AddRule` method adds a document integrity rule to a `SetupNode` object. For example, you can use this method to add a rule that specifies that a document type must contain a specific page type.
- [AddVariable method](#)
The `AddVariable` method adds a variable to a `SetupNode` object and assigns a default value.
- [DeleteRule method](#)
The `DeleteRule` method deletes the specified document integrity rule from a `SetupNode` object.
- [DeleteVariable method](#)
The `DeleteVariable` method deletes the specified variable that is referenced by an index, from a setup node.
- [DeleteVariableByName method](#)
The `DeleteVariableByName` method deletes the specified variable that is referenced by name from a setup node.
- [FindRule method](#)
The `FindRule` method uses a specified name to search for a `SetupNode` child object.
- [get_RuleChildName method](#)
The `get_RuleChildName` method gets the name of the rule under the current `SetupNode` object.
- [get_RuleMaxNumber Method](#)
The `get_RuleMaxNumber` method gets the maximum number of child objects of the specified type that can be associated with the parent object without violating a document integrity rule.
- [get_RuleMinNumber method](#)
The `get_RuleMinNumber` method gets the minimum number of child objects of the specified type that must be associated with the parent object to avoid violating a document integrity rule.
- [get_RuleObjectType method](#)
The `get_RuleObjectType` method gets the type of an object.
- [get_RulePosition method](#)
The `get_RulePosition` method gets the `pos` attribute of a specified rule within a `SetupNode` object. The `pos` attribute determines the position of an object type relative to other objects of the same type within the parent node. You can use this method to verify the order of pages in a document and ensure document integrity.
- [get_Variable method](#)
The `get_Variable` method gets the value of a variable from the `SetupNode` object. You can use this method to identify objects (for example, pages) that require special processing when a variable contains a specific value.
- [get_VariableName method](#)
The `get_VariableName` method gets the name of an indexed variable from the `SetupNode` object. You can use this method to identify objects (for example, pages) that require special processing when a variable contains a specific value. This method is useful when a node contains many child objects and variables.
- [get_VariableValue Method](#)
The `get_VariableValue` method gets the value of an indexed variable from the `SetupNode` object. You can use this method to identify objects (for example, pages) that require special processing when a variable contains a specific value. This method is useful when a node contains many child objects and variables.
- [GetRule method](#)
The `GetRule` method gets the `SetupNode` object that is referenced by a specified rule. You can use this method to identify objects (for example, pages) that require special processing when a node contains many child objects and rules.
- [NumOfRules method](#)
The `NumOfRules` method returns the number of rules that is within a `SetupNode` object.
- [NumOfVariables Method](#)
The `NumOfVariables` method returns the number of variables that is within a `SetupNode` object.

- [set_RuleChildName](#)
The `set_RuleChildName` method sets the name of the rule under the current `SetupNode` object.
- [set_RuleMaxNumber](#)
The `set_RuleMaxNumber` method sets the maximum number of child objects of the specified type that can be associated with the parent object without violating a document integrity rule.
- [set_RuleMinNumber](#)
The `set_RuleMinNumber` method sets the minimum number of child objects of the specified type that must be associated with the parent object to avoid violating a document integrity rule.
- [set_RuleObjectType](#)
The `set_RuleObjectType` method sets the type of an object.
- [set_RulePosition](#)
The `set_RulePosition` method sets the `pos` attribute of a specified rule within a `SetupNode` object. The `pos` attribute determines the position of an object type relative to other objects of the same type within the parent node. You can use this method to determine the order of pages in a document and ensure document integrity.
- [set_Variable](#)
The `set_Variable` method sets the value of a variable from the `SetupNode` object. You can use this method to determine which objects (for example, pages) require special processing when a variable contains a specific value.
- [set_VariableName](#)
The `set_VariableName` method sets the name of an indexed variable within a `SetupNode` object. You can use this method to identify variables within a node that contains many child objects.
- [set_VariableValue](#)
The `set_VariableValue` method sets the value of an indexed variable from the `SetupNode` object. You can use this method to identify objects (for example, pages) when a node contains many child objects and variables.

Parent topic: [DCOSetupNode APIs](#)

AddRule method

The `AddRule` method adds a document integrity rule to a `SetupNode` object. For example, you can use this method to add a rule that specifies that a document type must contain a specific page type.

Document integrity rules specify the required structure of the document. For example, a rule for the document type `Invoice` might specify that documents of this type must have a page of type `Main_Page`. A rule for the page type `Main_Page` might specify that pages of this type must have a field of type `Invoice_Total`.

If a `SetupNode` object for the specified child object does not exist, this method creates the `SetupNode` object.

Syntax

VBScript

```
AddRule(ObjectType as Long, lpszChildName as String,
         nPosition as Long, MinNumber as Long, MaxNumber as Long)
         as Boolean
```

C#

```
int AddRule(int nObjectType, string lpszChildName,
            short nPosition, short nMinNumber, short nMaxNumber)
```

Arguments

nObjectType

Use one of the following number values to determine the object type of the child node:

0 = Batch

1 = Document

2 = Page

3 = Field

lpszChildName

The name of the child object, which is stored as the `type` attribute of the rule.

nPosition

The `position` that child objects of this type must be relative to other child items at run time. In the DCO Setup window, this value is the `Order` property of the child node.

nMinNumber

The minimum number of instances of this child object that must exist for the document structure to be valid.

nMaxNumber

The maximum number of instances of this child object that can exist for the document structure to be valid.

Returns

Returns true if successful; returns false if unsuccessful.

C# example

This example gets the `SetupNode` object for the Invoice document in the Setup DCO. The example then adds a rule that requires one instance of the page `NewPage` to exist for the document structure to be valid. If a page node for `NewPage` does not exist, the method creates `NewPage` automatically.

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
TDCOLib.DCOSetup m_oDCOSetup = m_oDCO.SetupObject();
TDCOLib.DCOSetupNode m_oDCOSetupNode =
    m_oDCOSetup.GetNodeByName(1, "Invoice");
m_oDCOSetupNode.AddRule(2, "NewPage", 0, 1, 1);
```

The following line is added to the Setup DCO:

```
<D type="Invoice">
.
.
.
<P type="NewPage" pos="0" min="1" max="1"/> <!-- New line added -->
```

Additionally, if `NewPage` does not exist, the following page node is created:

```
<P type="NewPage">
    <V n="ID">0</V>
    <V n="TYPE">Page</V>
    <V n="STATUS">0</V>
    <V n="IMAGEFILE"></V>
    <V n="DATAFILE"></V>
    <V n="TEMPLATE IMAGE"></V>
    <V n="MIN_TYPES">0</V>
    <V n="MAX_TYPES">0</V>
</P>
```

Parent topic: [DCOSetupNode methods](#)

Related reference:

[DeleteRule method](#)

AddVariable method

The AddVariable method adds a variable to a SetupNode object and assigns a default value.

By default, all SetupNode objects include the standard variables that are installed with Datacap. For more information, see the *Standard Variable Reference* section in IBM® Knowledge Center for Datacap.

Syntax

VBScript

```
oSNO.AddVariable(lpszName as String, lpszValue as String) as Boolean
```

C#

```
bool AddVariable(string lpszName, string lpszValue)
```

Applies to

All node types.

Arguments

lpszName

Name of the new variable.

lpszValue

Default value (always a string)

Returns

Returns true if successful; returns false if unsuccessful.

C# example

This example gets the SetupNode object for the Invoice document in the APT.XML Setup DCO file and adds a variable *NewVar* with a default value *True*:

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");  
TDCOLib.DCOSetup m_oDCOSetup = m_oDCO.SetupObject();  
TDCOLib.DCOSetupNode m_oDCOSetupNode =  
    m_oDCOSetup.GetNodeByName(1, "Invoice");  
m_oDCOSetupNode.AddVariable("NewVar", "True");
```

The following line is added to the Setup DCO:

```
<D type="Invoice">  
    .  
    .  
    .  
<V n="NewVar">True</V> <!-- New line added -->
```

Parent topic: [DCOSetupNode methods](#)

Related information:

DeleteRule method

The `DeleteRule` method deletes the specified document integrity rule from a `SetupNode` object.

Description

For a description of document integrity rules, see [AddRule method](#).

Syntax

VBScript

```
oSNO.DeleteRule(nIndex as Long) as Boolean
```

C#

```
bool DeleteRule(int nIndex)
```

Arguments

`nIndex`

The index of the rule to delete, where 0 is the first rule that is specified under this `SetupNode` object.

Returns

True if successful; False otherwise.

See also

[AddRule method](#)

Parent topic: [DCOSetupNode methods](#)

DeleteVariable method

The `DeleteVariable` method deletes the specified variable that is referenced by an index, from a setup node.

Syntax

VBScript

```
oSNO.DeleteVariable(VarIndex as Long) as Boolean
```

C#

```
bool DeleteVariable(int nIndex)
```

Applies to

All node types.

Arguments

VarIndex

The index of the variable to delete, where 0 is the first variable.

Returns

True if successful; False otherwise.

See also

[AddVariable method](#), [DeleteVariableByName method](#)

Parent topic: [DCOSetupNode methods](#)

DeleteVariableByName method

The `DeleteVariableByName` method deletes the specified variable that is referenced by name from a setup node.

Syntax

VBScript

```
oSNO.DeleteVariableByName(lpszName as String) as Boolean
```

C#

```
bool DeleteVariableByName(string lpszName)
```

Applies to

All node types

Arguments

lpszName

The variable name

Returns

True if successful; False otherwise.

See also

[AddVariable method](#), [DeleteVariable method](#)

Parent topic: [DCOSetupNode methods](#)

FindRule method

The `FindRule` method uses a specified name to search for a `SetupNode` child object.

Syntax

VBScript

```
oSNO.FindRule(lpszName as String) child node as node object
```

C#

```
TDCOLib.DCOSetupNode FindRule(string lpszName)
```

Arguments

lpszName
The name of the rule.

Returns

The child node if successful; or nothing (null) otherwise.

Applies to

Any SetupNode object

See also

[GetRule method](#)

Parent topic: [DCOSetupNode methods](#)

get_RuleChildName method

The `get_RuleChildName` method gets the name of the rule under the current SetupNode object.

Attention: .NET only. For VBScript, use the [RuleChildName](#) property instead.

Description

The name of the rule is the `type` attribute in the Setup DCO XML file, and is the same name as the child node to which the rule applies. In the following rule for documents of type `Invoice`, the name of the rule is `Main_Page`.

```
<D type="Invoice">  
  .  
  .  
  <P type="Main_Page" pos="0" min="1" max="1"/>    <!--Rule-->
```

Syntax

C#

```
string get_RuleChildName(int nIndex)
```

Arguments

nIndex
The index of the rule, where 0 is the first rule under this SetupNode object.

C# Example

This example gets the name of the first rule under the current SetupNode object:

```
string strRuleName = m_oDCOSetupNode.get_RuleChildName(0);
```

See also

[set_RuleChildName](#)

Parent topic: [DCOSetupNode methods](#)

get_RuleMaxNumber Method

The `get_RuleMaxNumber` method gets the maximum number of child objects of the specified type that can be associated with the parent object without violating a document integrity rule.

Attention: .NET only. For VBScript, use the [RuleMaxNum](#) property instead.

Description

The maximum number is the `max` attribute in the Setup DCO XML file. In the following rule, the maximum number of pages of type `Main_Page` that a document of type `Invoice` can have is 1.

```
<D type="Invoice">
  .
  .
  <P type="Main_Page" pos="0" min="1" max="1"/>    <!--Rule-->
```

Syntax

C#

```
int get_RuleMaxNumber(int nIndex)
```

Arguments

nIndex

The index of the rule, where 0 is the first rule under the current SetupNode object.

C# Example

This example gets the `max` attribute of the first rule under the current SetupNode object:

```
int nMaxNum = m_oDCOSetupNode.get_RuleMaxNumber(0);
```

See also

[set_RuleMaxNumber](#)

Parent topic: [DCOSetupNode methods](#)

get_RuleMinNumber method

The `get_RuleMinNumber` method gets the minimum number of child objects of the specified type that must be associated with the parent object to avoid violating a document integrity rule.

Attention: .NET only. For VBScript, use the [RuleMinNum](#) property instead.

Description

The minimum number is the `min` attribute in the Setup DCO XML file. In the following rule, the minimum number of pages of type `Main_Page` that a document of type `Invoice` must have is 1.

```
<D type="Invoice">
  .
  .
  <P type="Main_Page" pos="0" min="1" max="1"/>    <!--Rule-->
```

Syntax

C#

```
int get_RuleMinNumber(int nIndex)
```

Arguments

nIndex

The index of the rule index, where 0 is the first rule under the current `SetupNode` object.

C# example

This example gets the `min` attribute of the first rule under the current `SetupNode` object.

```
int nMinNum = m_oDCOSetupNode.get_RuleMinNumber(0);
```

See also

[set_RuleMinNumber](#)

Parent topic: [DCOSetupNode methods](#)

get_RuleObjectType method

The `get_RuleObjectType` method gets the type of an object.

Attention: .NET only. For VBScript, use the [RuleObjectType](#) property instead.

Description

The type of an object is defined by a value range of 0-3.

0 = Batch

1 = Document

2 = Page

3 = Field

In the following rule for documents of type `Invoice`, the object type is 2 (Page).

```
<D type="Invoice">
  .
  .
  <P type="Main_Page" pos="0" min="1" max="1"/>    <!--Rule-->
```

Syntax

C#

```
int get_RuleObjectType(int nIndex)
```

Arguments

nIndex

The index of the rule index, where 0 is the first rule under the current SetupNode object.

C# Example

This example gets the object type for the first rule under the current SetupNode object.

```
int nRuleType = m_oDCOSetupNode.get_RuleObjectType(0);
```

See also

[set_RuleObjectType](#)

Parent topic: [DCOSetupNode methods](#)

get_RulePosition method

The `get_RulePosition` method gets the `pos` attribute of a specified rule within a SetupNode object. The `pos` attribute determines the position of an object type relative to other objects of the same type within the parent node. You can use this method to verify the order of pages in a document and ensure document integrity.

Attention: .NET only. For VBScript, use the [RulePosition](#) property instead.

Description

The following rules specify position requirements for documents of type `Tax Return`.

- A `TaxSep` page must come before the `1040EZ` page and any `Attachment` pages.
- The `1040EZ` page must come before any `Attachment` pages.

```
<D type="Tax Return">
.
.
  <P type="TaxSep" pos="1" min="0" max="1"/>           <!--Rule 0-->
  <P type="1040EZ" pos="2" min="1" max="1"/>         <!--Rule 1-->
  <P type="Attachment" pos="3" min="0" max="0"/>     <!--Rule 2-->
```

If a rule has `pos="0"`, then the corresponding object type is not included in the process that confirms document integrity. For details about document integrity, see [DCOSetupNode APIs](#).

Syntax

C#

```
int get_RulePosition(int nIndex)
```

Arguments

nIndex

The index of the rule, where 0 is the first rule under the current SetupNode object.

C# Example

This example gets the position attribute for the first rule under the current SetupNode object.

```
int nPosValue = m_oDCOSetupNode.get_RulePosition(0);
```

See also

[set_RulePosition](#)

Parent topic: [DCOSetupNode methods](#)

get_Variable method

The `get_Variable` method gets the value of a variable from the SetupNode object. You can use this method to identify objects (for example, pages) that require special processing when a variable contains a specific value.

Attention: .NET only. For VBScript, use the [Variable property](#) property instead.

Description

The following example shows three variables that are associated with the page node `Main_Page` in the Setup DCO file.

```
<P type="Main_Page">
  <V n="ID">0</V>           0
  <V n="TYPE">Page</V>      1
  <V n="STATUS">0</V>      2
```

0 Variable 0 (name = ID, value = 0)

1 Variable 1 (name = TYPE, value = Page)

2 Variable 2 (name = STATUS, value = 0)

Syntax

C#

```
string get_Variable(string lpszName)
```

Arguments

lpszName

The variable name (case sensitive)

C# Example

The following example gets the value of the variable `TYPE` from the page node `Main_Page` in the Setup DCO.

```
TDCOLib.DCOSetupNode m_oDCOSetupNode = m_oDCOSetup.GetNodeByName(2, "Main_Page");
string strValue = m_oDCOSetupNode.get_Variable("TYPE");
```

See also

[get_VariableName method](#), [get_VariableValue Method](#), [set_Variable](#)

Parent topic: [DCOSetupNode methods](#)

get_VariableName method

The `get_VariableName` method gets the name of an indexed variable from the `SetupNode` object. You can use this method to identify objects (for example, pages) that require special processing when a variable contains a specific value. This method is useful when a node contains many child objects and variables.

Attention: .NET only. For VBScript, use the [VariableName](#) property instead.

Description

The following example shows three variables that are associated with the page node `Main_Page` in the Setup DCO file:

```
<P type="Main_Page">
  <V n="ID">0</V>           0
  <V n="TYPE">Page</V>      1
  <V n="STATUS">0</V>      2
```

0 Variable 0 (name = ID, value = 0)

1 Variable 1 (name = TYPE, value = Page)

2 Variable 2 (name = STATUS, value = 0)

Syntax

C#

```
string get_VariableName(int nIndex)
```

Arguments

`nIndex`

The index of the variable, where 0 is the first variable that is defined in the `SetupNode` object

C# Example

The following example gets the name of the first variable that is defined within the page node, `Main_Page`, in the Setup DCO.

```
TDCOLib.DCOSetupNode m_oDCOSetupNode = m_oDCOSetup.GetNodeByName(2, "Main_Page");
string varName = m_oDCOSetupNode.get_VariableName(0);
```

See also

[ReadSetup method](#), [GetNodeByName method](#), [get_Variable method](#), [get_VariableValue Method](#), [set_VariableName](#)

Parent topic: [DCOSetupNode methods](#)

get_VariableValue Method

The `get_VariableValue` method gets the value of an indexed variable from the `SetupNode` object. You can use this method to identify objects (for example, pages) that require special processing when a variable contains a specific value. This method is useful when a node contains many child objects and variables.

Attention: .NET only. For VBScript, use the [VariableValue](#) property instead.

Description

The following example shows three variables that are associated with the page node, `Main_Page`, in the Setup DCO file:

```
<P type="Main_Page">
  <V n="ID">0</V>          0
  <V n="TYPE">Page</V>    1
  <V n="STATUS">0</V>     2
```

0 Variable 0 (name = ID, value = 0)

1 Variable 1 (name = TYPE, value = Page)

2 Variable 2 (name = STATUS, value = 0)

Syntax

C#

```
string get_VariableValue(int nIndex)
```

Arguments

nIndex

The index of the variable, where 0 is the first variable defined in the `SetupNode` object

C# Example

The following example gets the value of the first variable defined with the page node, `Main_Page`, in the Setup DCO:

```
TDCOLib.DCOSetupNode m_oDCOSetupNode = m_oDCOSetup.GetNodeByName(2, "Main_Page");
string varValue = m_oDCOSetupNode.get_VariableValue(0);
```

See also

[get_Variable](#) method, [get_VariableName](#) method, [set_VariableValue](#)

Parent topic: [DCOSetupNode methods](#)

GetRule method

The `GetRule` method gets the `SetupNode` object that is referenced by a specified rule. You can use this method to identify objects (for example, pages) that require special processing when a node contains many child objects and rules.

Description

For example, the following rule references a page of type `Main_Page`. So, `GetRule` returns the `Main_Page SetupNode` object.

```
<D type="Invoice">
  .
  .
  <P type="Main_Page" pos="0" min="1" max="1"/>      <!--Rule referencing page
"Main_Page" -->
```

Syntax

VBScript

```
oSNO.GetRule(nIndex as Long) child node as node object
```

C#

```
TDCOLib.DCOSetupNode GetRule(int nIndex)
```

Arguments

`nIndex`

The index of the rule, where 0 is the first rule under this `SetupNode` object.

Applies to

All node types

Returns

Child node object if successful; otherwise nothing (null).

C# Example

This example gets the `SetupNode` object for the `Invoice` document in the Setup DCO and then gets the first rule under this node. It then gets the `SetupNode` object for the child object that is referenced by this rule.

```
m_oDCO.ReadSetup("C:\\Datacap\\APT\\dco_APT\\APT.XML");
TDCOLib.DCOSetup m_oDCOSetup = m_oDCO.SetupObject();
TDCOLib.DCOSetupNode m_oDCOSetupNode = m_oDCOSetup.GetNodeByName(1, "Invoice");
TDCOLib.DCOSetupNode m_oDCOSetupNode2 = m_oDCOSetupNode.GetRule(0);
```

See also

[AddRule method](#)

Parent topic: [DCOSetupNode methods](#)

NumOfRules method

The `NumOfRules` method returns the number of rules that is within a `SetupNode` object.

Syntax

VBScript

```
oSNO.NumOfRules ( ) as Long
```

C#

```
int NumOfRules ()
```

Arguments

None

Returns

The number of rules in this SetupNode object.

Applies to

All node types

C# Example

This example returns the number of rules that is contained in the current SetupNode object.

```
int nNumRules = m_oDCOSetupNode.NumOfRules ();
```

See also

[AddRule method](#)

Parent topic: [DCOSetupNode methods](#)

NumOfVariables Method

The `NumOfVariables` method returns the number of variables that is within a SetupNode object.

Description

This method returns user-defined variables and standard variables. (For more information, see the *Standard Variable Reference*).

Syntax

VBScript

```
oSNO.NumOfVariables ( ) as Long
```

C#

```
int NumOfVariables ()
```

Arguments

None

Returns

The number of variables in this SetupNode object.

C# Example

The following example returns the number of rules within the current SetupNode object:

```
int nNumVars = m_oDCOSetupNode.NumOfVariables();
```

Parent topic: [DCOSetupNode methods](#)

set_RuleChildName

The `set_RuleChildName` method sets the name of the rule under the current SetupNode object.

Description

Important: .NET only. For VBScript, use the [RuleChildName](#) property instead. For more information, see [get_RuleChildName](#).

Syntax

C#

```
void set_RuleChildName(int nIndex, string pVal)
```

Arguments

nIndex

The index of the rule, where 0 is the first rule under this SetupNode object.

See also

[get_RuleChildName](#)

Parent topic: [DCOSetupNode methods](#)

set_RuleMaxNumber

The `set_RuleMaxNumber` method sets the maximum number of child objects of the specified type that can be associated with the parent object without violating a document integrity rule.

Description

Important: .NET only. For VBScript, use the [RuleMaxNum](#) property instead. For more information, see [get_RuleMaxNumber](#).

Syntax

C#

```
void set_RuleMaxNumber(int nIndex, int pVal)
```

Arguments

- nIndex
The index of the rule, where 0 is the first rule under this SetupNode object.
- pVal
The maximum number of child objects.

See also

[get_RuleMaxNumber](#)

Parent topic: [DCOSetupNode methods](#)

set_RuleMinNumber

The `set_RuleMinNumber` method sets the minimum number of child objects of the specified type that must be associated with the parent object to avoid violating a document integrity rule.

Description

Important: .NET only. For VBScript, use the [RuleMinNum](#) property instead.

For more information, see [get_RuleMinNumber](#).

Syntax

```
C#  
  
void set_RuleMinNumber(int nIndex, int pVal)
```

Arguments

- nIndex
The index of the rule, where 0 is the first rule under the current SetupNode object.
- pVal
The minimum number of child objects.

See also

[get_RuleMinNumber](#)

Parent topic: [DCOSetupNode methods](#)

set_RuleObjectType

The `set_RuleObjectType` method sets the type of an object.

Description

Important: .NET only. For VBScript, use the [RuleObjectType](#) property instead.

For more information, see [get_RuleObjectType](#).

Syntax

C#

```
void set_RuleObjectType(int nIndex, int pVal)
```

Arguments

nIndex

The index of the rule, where 0 is the first rule under this SetupNode object.

pVal

The object type:

0 = Batch

1 = Document

2 = Page

3 = Field

See also

[get_RuleObjectType](#)

Parent topic: [DCOSetupNode methods](#)

set_RulePosition

The `set_RulePosition` method sets the `pos` attribute of a specified rule within a SetupNode object. The `pos` attribute determines the position of an object type relative to other objects of the same type within the parent node. You can use this method to determine the order of pages in a document and ensure document integrity.

Description

Important: .NET only. For VBScript, use the [RulePosition](#) property instead.

For more information, see [get_RulePosition](#).

Syntax

C#

```
void set_RulePosition(int nIndex, int pVal)
```

Arguments

nIndex

The index of the rule, where 0 is the first rule under this SetupNode object.

pVal

The position of the rule.

See also

[get_RulePosition](#)

Parent topic: [DCOSetupNode methods](#)

set_Variable

The `set_Variable` method sets the value of a variable from the `SetupNode` object. You can use this method to determine which objects (for example, pages) require special processing when a variable contains a specific value.

Description

Important: .NET only. For VBScript, use the [Variable](#) property instead.
For more information, see [get_Variable](#).

Syntax

C#

```
void set_Variable(string lpszName, string pVal)
```

Arguments

`lpszName`
The variable name (case sensitive).

`pVal`
The value (always a string).

See also

[get_Variable](#)

Parent topic: [DCOSetupNode methods](#)

set_VariableName

The `set_VariableName` method sets the name of an indexed variable within a `SetupNode` object. You can use this method to identify variables within a node that contains many child objects.

Description

Important: .NET only. For VBScript, use the [VariableName](#) property instead.

Syntax

C#

```
void set_VariableName(int nIndex, string pVal)
```

Arguments

`nIndex`
The index of the variable, where 0 is the first variable that is defined in the `SetupNode` object.

`pVal`
The variable name.

See also

[get_VariableName](#)

Parent topic: [DCOSetupNode methods](#)

set_VariableValue

The `set_VariableValue` method sets the value of an indexed variable from the `SetupNode` object. You can use this method to identify objects (for example, pages) when a node contains many child objects and variables.

Description

Important: .NET only. For VBScript, use the [VariableValue](#) property instead.

Syntax

C#

```
void set_VariableValue(int nIndex, string pVal)
```

Arguments

nIndex

The index of the variable, where 0 is the first variable that is defined in the `SetupNode` object.

pVal

The variable value (always a string).

See also

[get_VariableValue](#)

Parent topic: [DCOSetupNode methods](#)

Integrating with other products

Datacap integrates with other IBM enterprise content management solutions to enhance your capture applications.

- [Transactional capture](#)
Using Datacap Transactional capture, you can scan documents directly in IBM® Case Manager and IBM Content Navigator.

Transactional capture for IBM Content Navigator and IBM Case Manager

Use Datacap Transactional Capture to scan documents directly in IBM® Content Navigator and IBM Case Manager. You can also add documents to a Datacap batch from an IBM Content Navigator repository.

When you scan a document, you specify a document class and document properties, and then upload the document to a repository. You can also configure Datacap Transactional Capture to automatically populate document properties from the scanned document.

- [Sample workflows](#)
Transactional capture sample workflows show how you can access Datacap functionality in IBM Content Navigator and IBM Case Manager to scan documents and add documents to batches.

- [Configuring Datacap for transactional capture](#)
Use configuration steps in IBM Datacap to use transactional capture in IBM Content Navigator and IBM Case Manager.
- [Configuring scanning in IBM Content Navigator](#)
Configure IBM Content Navigator to enable scanning. Optionally, you can also enable automatic data capture.
- [Configuring scanning in IBM Case Manager](#)
You can configure IBM Case Manager to enable scanning. Optionally, you can also enable automatic data capture.
- [Configuring IBM Content Navigator for adding documents to batches](#)
Configuration steps are required to add documents in IBM Content Navigator to Datacap batches.

Sample workflows

Transactional capture sample workflows show how you can access Datacap functionality in IBM® Content Navigator and IBM Case Manager to scan documents and add documents to batches.

- [Configuring your workstation for scanning](#)
The first time that you scan in Case Manager Client or IBM Content Navigator, you might be prompted to download and run a program called DynamicWebTwainHTML5Edition.exe. You must install Web TWAIN scanning software on your workstation before you can scan. This is done one time only.
- [Scanning documents in IBM Content Navigator](#)
In this sample workflow, you scan a document in IBM Content Navigator by using the Scan Document or Add Document actions. This feature is available when you configure IBM Datacap and IBM Content Navigator for transactional capture.
- [Scanning documents in IBM Case Manager](#)
In this sample workflow, you scan a document in IBM Case Manager by selecting the Add Document from Scanner option. This feature is available when you configure IBM Datacap and IBM Case Manager for transactional capture.
- [Adding documents to a batch from an IBM Content Navigator repository](#)
In this sample workflow, you add a document in an IBM Content Navigator repository to a Datacap batch by clicking Add to Batch. Then you verify and submit the batch by using Datacap Navigator. This feature is available when you configure IBM Datacap and IBM Content Navigator for transactional capture.

Parent topic: [Transactional capture for IBM Content Navigator and IBM Case Manager](#)

Configuring your workstation for scanning

The first time that you scan in Case Manager Client or IBM® Content Navigator, you might be prompted to download and run a program called DynamicWebTwainHTML5Edition.exe. You must install Web TWAIN scanning software on your workstation before you can scan. This is done one time only.

Procedure

To install Web TWAIN scanning software:

1. When prompted, click the link on the Information window to download and save the DynamicWebTwainHTML5Edition.exe file.
2. Close the Information window.
3. Run the file to install the software on your workstation.

Parent topic: [Sample workflows](#)

Scanning documents in IBM Content Navigator

In this sample workflow, you scan a document in IBM® Content Navigator by using the Scan Document or Add Document actions. This feature is available when you configure IBM Datacap and IBM Content Navigator for transactional capture.

Procedure

To scan document in IBM Content Navigator:

1. Log in to an IBM Content Navigator desktop that contains a custom content list menu with the Add Document action for transactional capture.
2. Click Scan Document or Add Document to scan documents.
3. In the Save in field, select a location in your repository where you want to save the scanned document.
4. In the What do you want to save? field, select Document from scanner.
5. Select your scanner from the drop-down list, adjust scanner settings if necessary, load the document in the scanner, click Scan, and wait until the pages are scanned.
If want to import images from the repository, select Import from Directory and click Scan. Scanned images are displayed in a thumbnail strip. Double-click an image to view a full page.
6. Select a document class from the Class drop-down list.
7. If you configured transactional capture for automatic data capture, click Capture to populate properties of the document class. If SSO is not enabled, you are prompted to log in to the Datacap repository. You can specify whether to capture data from the current page or from all pages. Only matched field values are populated in the document properties.
8. Click Add to save the document.

Parent topic: [Sample workflows](#)

Scanning documents in IBM Case Manager

In this sample workflow, you scan a document in IBM® Case Manager by selecting the Add Document from Scanner option. This feature is available when you configure IBM Datacap and IBM Case Manager for transactional capture.

Procedure

To scan a document in IBM Case Manager:

1. Log in to Case Manager Client.
2. Double-click a case to open it.
3. On the Document tab, click the Add button and select Add Document from Scanner.
4. Select your scanner from the drop-down list. If you do not have a scanner, or if you want to import images instead, select Import from Directory.
5. Load the document in the scanner, click Scan > Append, and wait until the pages are scanned.
6. Images are scanned into the thumbnail strip. Double-click the first image to view a full page.
7. Select a document class from the drop-down list and fill in the document properties. If you configured transactional capture for automatic data capture, click Capture to populate properties of the document class. If SSO is not enabled, you are prompted to log in to the Datacap repository. You can specify whether to capture data from the current page or from all pages. Only matched field values are populated in the document properties.
8. Click Add to save the document in the case folder.

Parent topic: [Sample workflows](#)

Adding documents to a batch from an IBM Content Navigator repository

In this sample workflow, you add a document in an IBM® Content Navigator repository to a Datacap batch by clicking Add to Batch. Then you verify and submit the batch by using Datacap Navigator. This feature is available when you configure IBM Datacap and IBM Content Navigator for transactional capture.

Procedure

To add a document to a batch:

1. Log in to IBM Content Navigator and browse to a document in the repository.
2. Select the document to view the document properties. The class name should match the document type in the Setup DCO file of your Datacap application.
3. Click Add to Batch. A window is displayed with the new Datacap batch number.
4. Log in to Datacap Navigator and view the new batch in Job Monitor. Verify and submit the batch.

Parent topic: [Sample workflows](#)

Configuring Datacap for transactional capture

Use configuration steps in IBM® Datacap to use transactional capture in IBM Content Navigator and IBM Case Manager.

Procedure

To configure Datacap for transactional capture, complete the following steps:

1. Deploy the Datacap Navigator plug-in in IBM Content Navigator. For instructions, see [Datacap Navigator installation steps](#).
2. Create a Datacap application for transactional capture. In the Datacap application Setup DCO, document types must have the same symbolic names as document classes that are defined in your IBM Content Navigator repository.
3. Optional: Configure transactional capture to capture data from documents. Captured data populates document fields in the field entry panel when you scan documents. Otherwise, you must manually enter data in document fields. Captured data also populates document properties when you add documents to a new Datacap batch.
 - a. Use Datacap Studio to create a task profile named TransactionCaptureOCR. For more information about using Datacap Studio, see [Quick tour of the user interface](#).
 - b. Under the TransactionCaptureOCR task profile, add rulesets that recognize data in scanned documents and generate page data files. For example, for the TravelDocs sample Datacap application, add the following rulesets:
 - ImageFix
 - PageID
 - CreateDocs
 - Document Integrity
 - Recognize
 - Clean
 - Validate

When you scan a document, transactional capture populates the corresponding document fields with the recognized data from the page data files.

- c. In your Datacap application, ensure that the page types contain fields that have the same symbolic names as the properties under the document class of your repository such as IBM FileNet® Content Manager or IBM Content Manager Enterprise Edition.
- 4. For adding documents to batches only: Ensure that the user who adds documents to batches has permissions to create batches in Datacap. For more information, see [Users, groups, and stations administration](#).
- 5. Optional: For adding documents to batches only: In an IBM Content Navigator repository, one document can contain multiple TIF files. You can create rulesets or a task profile to split the document into multiple TIF files, when you add documents to a batch. Otherwise, single document that contains multiple TIF files is added to the batch. Create rulesets to split a document into multiple TIF files:
 - a. In Datacap Studio, right-click the Convert Files To Images ruleset and select Install in application.
 - b. Select the following options:

Setting	Value
File naming pattern	Hierarchical
Source Page	<ul style="list-style-type: none"> ■ Remove page after processing ■ Multipage TIFF Expansion

- c. Click Save.
- d. Add the ruleset to the PageID profile.
- 6. Add your Datacap application as a Datacap repository in IBM Content Navigator. For more information, see [Adding applications to Datacap Navigator](#).
- 7. Add the Datacap application to your desktop by using the IBM Content Navigator administration tool.
- 8. Enable SSO between Datacap and IBM Content Navigator repositories. If you do not enable SSO, you are prompted to log on to the Datacap repository when you scan documents and add documents to batches in IBM Content Navigator and IBM Case Manager. For more information, see [Configuring single sign-on \(SSO\) for Datacap Navigator](#).

Parent topic: [Transactional capture for IBM Content Navigator and IBM Case Manager](#)

Configuring scanning in IBM Content Navigator

Configure IBM® Content Navigator to enable scanning. Optionally, you can also enable automatic data capture.

Before you begin

Ensure that Datacap is configured for transactional capture. For instructions, see [Configuring Datacap for transactional capture](#).

Procedure

To configure scanning in IBM Content Navigator:

1. Create a custom content list menu with the Add Document action by using the IBM Content Navigator administration tool.
 - a. Access the IBM Content Navigator administration tool. In a browser, enter a URL with the following format:


```
http://host_name:port_number/context_root/?desktop=admin
```

By default, the context root is `navigator`.
 - b. Click Menus in the left pane.
 - c. Create a custom content list that contains the Add Document action and the Scan Document action for transactional capture.

Important: Two Add Document actions are available for selection. To choose the correct action, place your mouse cursor over the Add Document actions to display the action properties. Select the Add Document action with the following property: Plug-in: DatacapWebPlugin.

d. Assign the custom content list to the content list toolbar of the desktop.

- Optional: Configure your IBM Content Navigator repository for automatic data capture. When configured, data captured from scanned documents populates document fields in the field entry panel. Document classes must have properties that are mapped to page level fields in the Datacap application setup DCO. For example, you can configure an IBM FileNet® Content Manager repository for automatic data capture with the TravelDocs sample Datacap application. The TravelDocs setup DCO contains a Car_Rental document type and a Pickup_Date field in the Rental_Agreement page type. To configure the repository, create a Car_Rental document class the repository, and create a Pickup_Date property for the Car_Rental document class.

Important: Specify the string data type for properties in your repository.

Parent topic: [Transactional capture for IBM Content Navigator and IBM Case Manager](#)

Configuring scanning in IBM Case Manager

You can configure IBM® Case Manager to enable scanning. Optionally, you can also enable automatic data capture.

Before you begin

Ensure that Datacap is configured for transactional capture. For instructions, see [Configuring Datacap for transactional capture](#).

About this task

The following sample configuration adds an option called Add Document from Scanner to the Add button in Case Manager Client. Case workers can select the scan option among the other options for adding documents.

Procedure

To configure scanning in IBM Case Manager:

- Optional: Configure your IBM Case Manager repository for automatic data capture. When configured, data captured from scanned documents populates document fields in the field entry panel. Document classes must have properties that are mapped to page level fields in the Datacap application setup DCO. For example, you can configure an IBM FileNet® Content Manager repository for automatic data capture with the TravelDocs sample Datacap application. The TravelDocs setup DCO contains a Car_Rental document type, and a Pickup_Date field that is in the Rental_Agreement page type. To configure the repository, create a Car_Rental document class in the repository, and create a Pickup_Date property for the Car_Rental document class.
Important: Specify the string data type for properties in your repository.
- Log in to Case Manager Builder.
- On the Manage Solutions page, create a new solution or click Edit on the solution that you want to add transactional capture to.
- Optional: If you are configuring IBM Case Manager for automatic data capture, click the Properties tab, click Add PropertyReuse Property, and select all properties that you created for data capture.
- On the Pages tab, click the Open Page Designer icon for the Case Details page that is used for the solution.
- Click the Edit Settings icon for the Case Information widget.

7. Select a location for the scan function. For example, you can add the scan function to the Add button menu on the Documents view toolbar. Click the Toolbars tab, click Add Menu Item, and select Scan Document from the list of actions.
8. Enter a label for the menu item, such as Add Document from Scanner. Then, click OK.
9. The menu item is added to the list. Click OK at the bottom of the Case Information window to save your changes.
10. On the Page Designer page, click Save and then click Close to save your changes.
11. On the solution page, click Save and Close.
12. Click the Commit link on the solution.
13. Click the Deploy link on the solution.
14. Click Deploy in the Confirmation window, wait for the deployment to complete, log out of Case Manager Builder, and close the browser.

Parent topic: [Transactional capture for IBM Content Navigator and IBM Case Manager](#)

Configuring IBM Content Navigator for adding documents to batches

Configuration steps are required to add documents in IBM® Content Navigator to Datacap batches.

Before you begin

Ensure that Datacap is configured for transactional capture. For instructions, see [Configuring Datacap for transactional capture](#).

About this task

When you add a document to a new batch on Datacap Server from an IBM Content Navigator repository, a DCO document is created for each added document. The DCO document contains a page of type Other and the content of the added document. Document properties are saved as document variables in the batch.

Procedure

To configure IBM Content Navigator for adding documents to batches:

1. Access the IBM Content Navigator administration tool. In a browser, enter a URL with the following format:

```
http://host_name:port_number/context_root/?desktop=admin
```

By default, the context root is `navigator`.

2. Click Menus in the left pane.
3. Create a custom content list that contains the Add to Batch action.

Parent topic: [Transactional capture for IBM Content Navigator and IBM Case Manager](#)

Troubleshooting and support

When you experience problems, you might need to perform troubleshooting tasks to determine the corrective action to take.

- [Troubleshooting process overview](#)
Troubleshooting is a systematic approach to solving a problem. The goal of troubleshooting is to determine why something does not work as expected and how to resolve the problem. You can search live, Web-based support resources by using the Web search form under Searching knowledge bases in the left navigation pane.
- [Troubleshooting Datacap security and authentication](#)
You might encounter problems when you start the Datacap system or log on to a component. For example, in a client/server environment, you might see a problem with the automatic encryption key import feature.
- [Troubleshooting Rulerunner](#)
You might need to troubleshoot problems when you use Rulerunner to automatically process background tasks when batches are pending.
- [Troubleshooting FastDoc](#)
You might need to troubleshoot problems when you use FastDoc to scan, process, verify, and export documents.
- [Troubleshooting Datacap web services](#)
If the file upload fails on large files or the PUT request method endpoints are not working, you can check to ensure that your configurations are correct.
- [Log files](#)
To obtain adequate logging information for troubleshooting purposes, you must enable logging for the client and for the Rulerunner Service.

Troubleshooting process overview

Troubleshooting is a systematic approach to solving a problem. The goal of troubleshooting is to determine why something does not work as expected and how to resolve the problem. You can search live, Web-based support resources by using the Web search form under Searching knowledge bases in the left navigation pane.

The first step in the troubleshooting process is to describe the problem completely. Problem descriptions help you and the IBM Support representative know where to start to find the cause of the problem. This step includes asking yourself basic questions:

- What are the symptoms of the problem?
- Where does the problem occur?
- When does the problem occur?
- Under which conditions does the problem occur?
- Can the problem be reproduced?

The answers to these questions typically lead to a good description of the problem, which can then lead you a problem resolution.

What are the symptoms of the problem?

When starting to describe a problem, the most obvious question is "What is the problem?" This question might seem straightforward; however, you can break it down into several focused questions that create a more descriptive picture of the problem. These questions can include:

- Who, or what, is reporting the problem?
- What are the error codes and messages?
- How does the system fail? For example, is it a loop, hang, crash, performance degradation, or incorrect result?

Where does the problem occur?

Determining where the problem originates is not always easy, but it is one of the most important steps in resolving a problem. Many layers of technology can exist between the reporting and failing components. Networks, disks, and drivers are only a few of the components to consider when you are investigating problems.

The following questions help you to focus on where the problem occurs to isolate the problem layer:

- Is the problem specific to one platform or operating system, or is it common across multiple platforms or operating systems?
- Is the current environment and configuration supported?

If one layer reports the problem, the problem does not necessarily originate in that layer. Part of identifying where a problem originates is understanding the environment in which it exists. Take some time to completely describe the problem environment, including the operating system and version, all corresponding software and versions, and hardware information. Confirm that you are running within an environment that is a supported configuration; many problems can be traced back to incompatible levels of software that are not intended to run together or have not been fully tested together.

When does the problem occur?

Develop a detailed timeline of events leading up to a failure, especially for those cases that are one-time occurrences. You can most easily develop a timeline by working backward: Start at the time an error was reported (as precisely as possible, even down to the millisecond), and work backward through the available logs and information. Typically, you need to look only as far as the first suspicious event that you find in a diagnostic log.

To develop a detailed timeline of events, answer these questions:

- Does the problem happen only at a certain time of day or night?
- How often does the problem happen?
- What sequence of events leads up to the time that the problem is reported?
- Does the problem happen after an environment change, such as upgrading or installing software or hardware?

Responding to these types of questions can give you a frame of reference in which to investigate the problem.

Under which conditions does the problem occur?

Knowing which systems and applications are running at the time that a problem occurs is an important part of troubleshooting. These questions about your environment can help you to identify the root cause of the problem:

- Does the problem always occur when the same task is being performed?
- Does a certain sequence of events need to occur for the problem to surface?
- Do any other applications fail at the same time?

Answering these types of questions can help you explain the environment in which the problem occurs and correlate any dependencies. Remember that just because multiple problems might have occurred around the same time, the problems are not necessarily related.

Can the problem be reproduced?

From a troubleshooting standpoint, the ideal problem is one that can be reproduced. Typically, when a problem can be reproduced you have a larger set of tools or procedures at your disposal to help you investigate. Consequently, problems that you can reproduce are often easier to debug and solve. However, problems that

you can reproduce can have a disadvantage: If the problem is of significant business impact, you do not want it to recur. If possible, re-create the problem in a test or development environment, which typically offers you more flexibility and control during your investigation.

- Can the problem be re-created on a test system?
- Are multiple users or applications encountering the same type of problem?
- Can the problem be re-created by running a single command, a set of commands, or a particular application?
- [Searching knowledge bases](#)
You can often find solutions to problems by searching IBM knowledge bases. You can optimize your results by using available resources, support tools, and search methods.
- [Getting fixes from Fix Central](#)
You can use Fix Central to find the fixes that are recommended by IBM Support for a variety of products. With Fix Central, you can search, select, order, and download fixes for your system with a choice of delivery options.
- [Contacting IBM Software Support](#)
IBM® Software Support provides assistance with product defects, answering FAQs, and performing rediscovery.
- [Subscribing to Support updates](#)
To stay informed of important information about the IBM products that you use, you can subscribe to updates.

Searching knowledge bases

You can often find solutions to problems by searching IBM knowledge bases. You can optimize your results by using available resources, support tools, and search methods.

About this task

IBM® provides extensive documentation within Knowledge Center. You can find useful information by searching Knowledge Center, but sometimes you need to look beyond Knowledge Center to answer your questions or resolve problems.

Procedure

To search knowledge bases for information that you need, use one or more of the following approaches:

- Search for content by using the IBM Support Assistant (ISA). ISA is a no-charge software serviceability workbench that helps you answer questions and resolve problems with some IBM software products. You can find instructions for downloading and installing ISA on the [ISA website](#) (www.ibm.com/software/support/isa/). Some IBM products have not fully implemented support for ISA.
- Find the content that you need by using the IBM Support Portal: <http://www.ibm.com/supportportal> (www.ibm.com/supportportal).

The IBM Support Portal is a unified, centralized view of all technical support tools and information for all IBM systems, software, and services. The IBM Support Portal lets you access the IBM electronic support portfolio from one place. You can tailor the pages to focus on the information and resources that you need for problem prevention and faster problem resolution. Familiarize yourself with the IBM Support Portal by viewing the [demo videos](#) (https://www.ibm.com/blogs/SPNA/entry/the_ibm_support_portal_videos) about this tool. These videos introduce you to the IBM Support Portal, explore troubleshooting and other resources, and demonstrate how you can tailor the page by moving, adding, and deleting portlets.

- To search multiple Internet resources for your product, expand Searching knowledge bases in the navigation frame to the left and select Web search. From this topic, you can conduct a Google search or search within IBM developerWorks®. You can also link directly to the support portal for this product where you can search a variety of resources including:
 - IBM technotes
 - IBM downloads
 - IBM Redbooks®
- Search for content by using the IBM masthead search. You can use the IBM masthead search by typing your search string into the Search field at the top of any ibm.com® page.
- Search for content by using any external search engine, such as Google, Yahoo, or Bing. If you use an external search engine, your results are more likely to include information that is outside the ibm.com domain. However, sometimes you can find useful problem-solving information about IBM products in newsgroups, forums, and blogs that are not on ibm.com.
Tip: Include "IBM" and the name of the product in your search if you are looking for information about an IBM product.

Parent topic: [Troubleshooting process overview](#)

Related tasks:

[Contacting IBM Software Support](#)

Getting fixes from Fix Central

You can use Fix Central to find the fixes that are recommended by IBM Support for a variety of products. With Fix Central, you can search, select, order, and download fixes for your system with a choice of delivery options.

Procedure

To find and install fixes:

1. Go to [Fix Central](http://www.ibm.com/support/fixcentral) (www.ibm.com/support/fixcentral) to download the installation program for product fixes. Fix Central provides download, installation, and configuration instructions for the installation program.
2. Select Enterprise Content Management as your product group, then select your product, version and platform, and click Continue.
3. Browse or search for fixes based on criteria such as (Authorized Program Analysis Report) APAR number or fix ID, or just request a list of recommended fixes.
4. If you have already registered for My support, sign in and skip to the next step. If you have not registered, click Register now. Complete the registration form using your email address as your IBM ID and click Submit.
5. Select and download the fix using your preferred download method. When downloading the file, ensure that the name of the maintenance file is not changed. This change might be intentional, or it might be an inadvertent change that is caused by certain web browsers or download utilities.
6. Apply the fix.
7. Optional: Subscribe to receive weekly email notifications about fixes and other IBM Support updates.

Parent topic: [Troubleshooting process overview](#)

Related tasks:

[Subscribing to Support updates](#)

Contacting IBM Software Support

IBM® Software Support provides assistance with product defects, answering FAQs, and performing rediscovery.

About this task

After trying to find your answer or solution by using other self-help options such as technotes, you can contact IBM Software Support. Before you contact IBM Software Support, your company must have an active IBM software subscription and support contract, and you must be authorized to submit problems to IBM. The type of software subscription and support contract that you need depends on the type of product you have. For information about the types of available support, see the Support portfolio topic in the [Software Support Handbook](#).

Procedure

Complete the following steps to contact IBM Software Support with a problem:

1. Define the problem, gather background information, and determine the severity of the problem. To determine the severity level, you need to understand and assess the business impact of the problem you are reporting. Use the following criteria:

Severity	Business impact
1	Critical business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
2	Significant business impact: The program is usable but is severely limited.
3	Some business impact: The program is usable with less significant features (not critical to operations) unavailable.
4	Minimal business impact: The problem causes little impact on operations, or a reasonable circumvention to the problem has been implemented.

For more information, including examples of each severity level, see the "Getting IBM support" topic in the *IBM Software Support Handbook*.

2. Gather diagnostic information. Use the Web search form (under Troubleshooting and support > Searching knowledge bases in the navigation pane of this Knowledge Center) to search for the keyword `mustgather` to see whether there are specific files or other diagnostic information to gather.

For example:

- o What software versions were you running when the problem occurred?
- o Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- o Can the problem be re-created? If so, what steps led to the failure?
- o Have any changes been made to the system (for example, hardware, operating system, networking software, and so on)?
- o Are you currently using a workaround for this problem? If so, please be prepared to explain it when you report the problem.

3. Submit the problem to IBM Support.

IBM Support Assistant (ISA)

Go to the Software support site to use the [IBM Support Assistant](#) (www.ibm.com/software/support/isa/).

Online

Go to the [IBM Support Portal](#) (www.ibm.com/software/support/). You can open, update, and view all your Service Requests from the Service Request portlet on the Service Request page.

Phone

For the phone number to call in your country, see the [Directory of worldwide contacts](#) web page (www.ibm.com/planetwide/).

Results

If the problem that you submit is for a software defect or for missing or inaccurate documentation, IBM Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM Support website daily, so that other users who experience the same problem can benefit from the same resolution.

Parent topic: [Troubleshooting process overview](#)

Related tasks:

[Searching knowledge bases](#)

Related information:

[IBM Software Support Handbook](#)

Subscribing to Support updates

To stay informed of important information about the IBM products that you use, you can subscribe to updates.

About this task

By subscribing to receive updates, you can receive important technical information and updates for specific Support tools and resources. You can subscribe to updates by using one of two approaches:

RSS feeds and social media subscriptions

For general information about RSS, including steps for getting started and a list of RSS-enabled IBM web pages, visit the IBM Software Support RSS feeds site.

My Notifications

With My Notifications, you can subscribe to Support updates for any IBM product. You can specify that you want to receive daily or weekly email announcements. You can specify what type of information you want to receive (such as publications, hints and tips, product flashes (also known as alerts), downloads, and drivers). My Notifications enables you to customize and categorize the products about which you want to be informed and the delivery methods that best suit your needs.

Procedure

To subscribe to Support updates:

1. Go to the [IBM® Support Portal](#) and click Sign in to create, manage, or view your subscriptions in the Notifications pane.
2. If you have already registered for My support, sign in and skip to the next step. If you have not registered, click Register now. Complete the registration form using your email address as your IBM ID and click Submit.
3. Identify what and how you want to receive updates.
 - a. Click the Subscribe tab.
 - b. Click Information Management or another software brand or type of hardware.
 - c. Select one or more products by name, for example, Content Manager Enterprise Edition or FileNet® Content Manager, and click Continue.
 - d. Select your preferences for how to receive updates, whether by e-mail, online in a designated folder, or as an RSS or Atom feed.
 - e. Select the types of documentation updates you want to receive, for example, new information about product downloads and discussion group comments.
 - f. Click Submit.

Results

Until you modify your RSS feeds and My Notifications preferences, you receive notifications of updates that you have requested. You can modify your preferences when needed (for example, if you stop using one product and begin using another product).

Parent topic: [Troubleshooting process overview](#)

Related tasks:

[Getting fixes from Fix Central](#)

Related information:

[What is RSS?](#)

[IBM Software Support RSS feeds](#)

[Subscribe to My Notifications support content updates](#)

[My notifications for IBM Software support](#)

[My notifications for IBM Support overview](#)

Troubleshooting tips for IBM Datacap

This section provides a collection of troubleshooting tips for IBM Datacap.

- [Invalid character when scanning document by using Dynamic Web TWAIN HTML5 driver](#)
When scanning a document by using Dynamic Web TWAIN HTML5 driver, the response header shows an invalid character (*).
- [Scanner settings do not persist when scanning using Dynamic Web TWAIN Interface](#)
In Datacap Navigator, when scanning using the Dynamic Web TWAIN Interface, the scanner settings do not persist and sometimes also causes application crash.
- [After image rotation, snippet shows image in its original orientation](#)
In Datacap Navigator, when the image is rotated, snippet for the field shows image in its original orientation.
- [AppScan issues: Physical path disclosure and hidden directory detection](#)
When a user runs Taskmaster Web (tmWeb .Net) from the same machine it is installed on, in case of error 403 or error 404, the complete error page path or hidden directories are displayed .
- [Document integrity check fails to identify batch level variables “MIN TYPES” and “MAX TYPES”](#)
In the document hierarchy, at the batch level, document integrity check fails to identify batch level variables *MIN TYPES* and *MAX TYPES*.

Invalid character when scanning document by using Dynamic Web TWAIN HTML5 driver

When scanning a document by using Dynamic Web TWAIN HTML5 driver, the response header shows an invalid character (*).

Symptoms

When scanning a document using Dynamic Web TWAIN HTML5 driver, the response header shows an invalid character (*).

Example:

```
HTTP/1.1 200 OK
Connection:keep-alive
Server: dynamic_lws
Access-Control-Allow-Origin: *
Content-Type: text/json
Content-Length: 100
```

```
{
  "id" : "901931755",
  "method" : "IfAllowLocalCache",
  "result" : [ true ],
  "cmdId" : ""
}
```

Causes

The Dynamic Web TWAIN HTML5 Driver had a security vulnerability in which `Access-Control-Allow-Origin` in the response header was set to the wildcard character (*).

Resolving the problem

User response: The security vulnerability in Dynamic Web TWAIN HTML5 Driver was fixed by setting the `Access-Control-Allow-Origin` in response header to the authorized URL used to access the service.

To implement this fix, complete the following steps:

1. Before you connect a workstation to Datacap Navigator, uninstall the currently installed version of the Dynamic Web TWAIN HTML5 Driver by using the Programs and Features option in Microsoft Windows Control Panel.
2. The next time that you use the *Scan* task from Datacap Navigator Job Monitor, the system will prompt you to download *DynamicWebTWAINHTML5Edition.exe*.
3. Download this executable file and follow the installation wizard to complete the installation. Verify that the installed version is **12.2.7427**.
4. Now, proceed with the Scan task. You are prompted to authorize the operation by selecting one of the following options:
 - o Allow Once
 - o Always Allow
 - o Block Once
 - o Always Block

The *Allow Once* and *Block Once* options are only applied for the current session, whereas the *Always Allow* and *Always Block* options remain effective until changed by the user.

System administrator response: The following updated files are available in the directory [Datacap install folder]\tmweb.java.

- *DynamicWebTWAINHTML5Edition.msi*: For Microsoft Windows
- *DynamicWebTWAINHTML5MACFullEdition.pkg*: For Apple Mac OS

The system administrators must use the appropriate file to preinstall the updated Dynamic Web TWAIN HTML5 Driver on any scan workstation that connects to the updated Datacap Navigator server.

Note: Do not use any other MSI file in the [Datacap install folder]\tmweb.java directory.

Result:

After you install the updated Dynamic Web TWAIN HTML5 Driver and perform the document scan action, the response header displays an authorized URL in place of the wildcard character (*).

Example:

```
HTTP/1.1 200 OK
Connection:keep-alive
Server: dynamic_lws
Access-Control-Allow-Origin:<my_allowed-url>
```

```
Content-Type: text/json
Content-Length: 100
{
  "id" : "901931755",
  "method" : "IfAllowLocalCache",
  "result" : [ true ],
  "cmdId" : ""
}
```

Parent topic: [Troubleshooting tips for IBM Datacap](#)

Scanner settings do not persist when scanning using Dynamic Web TWAIN Interface

In Datacap Navigator, when scanning using the Dynamic Web TWAIN Interface, the scanner settings do not persist and sometimes also causes application crash.

Symptoms

In Datacap Navigator, when scanning using the Dynamic Web TWAIN Interface, the scanner settings do not persist. This sometimes causes the application to hang and crash.

Causes

Datacap applications prior to Version 9.1.1 did not support skipping the blank pages during scan process, hence there was no option to persist the scanner setting for skipping the blank pages using the Dynamic Web TWAIN API.

Resolving the problem

User response: You can now skip the blank pages during the scan process, by using the Advanced Task scanner settings. After you configure the Skip Blank Pages option, the setting is persisted.

Perform the following steps to resolve this issue in the Datacap Navigator.

1. Add the line of code (highlighted in the following code-snippet) as the last child element of the ScannerSet paragraph in your application scan task XML.

Note: For OOTB application *TravelDocs*, the file to be updated is `nscan.set.xml`

```
<V label="Enable Autofeeder" n="Autofeed" tip="Controls scanner source
- manual vs. autofeeder option" type="checkbox">1</V>
<V label="Use Duplex Mode" n="Duplex" tip="When enabled - scans both
sides of the image - duplex mode" type="checkbox">0</V>
<V label="Pixel Type" n="PixelFormat" tip="Specify pixel type for scanning
(0-BnW, 1-Grayscale, 2-RGB)" type="text">0</V>
<V label="Bit Depth" n="Bits" tip="Specify bitdepth for above image type"
type="text">1</V>
<V label="Image Resolution DPI" n="Resolution" tip="Specify image resolution
(X and Y resolution assumed the same) DPI" type="text">200</V>
<V label="Paper Size" n="PaperSize" tip="Specify paper size for scanning
(0-none, 1-A4, 2-JISB5)" type="text">0</V>
<V label="Image format" n="Extension"
tip="Specify extension for the scanned images: tif, jpeg, bmp, png"
type="text">tif</V>
<V label="Skip Blank Pages" n="SkipBlankPages"
tip="Specify whether to discard blank pages, default is do not discard.
```

For some scanners, duplex must be unchecked for this to be enabled."
`type="checkbox">0</v>`

2. Take a backup of DatacapWebPlugin.jar file before proceeding to update it, in case you need to revert the fix later.
3. Stop the IBM WebSphere Application Server service.
4. Replace the updated DatacapWebPlugin.jar in the Datacap installation folder.
Note: Example: C:\Datacap\tmweb.java
5. Start the IBM WebSphere Application Server service.
6. Log in to admin desktop.
Note: Example URL of Admin desktop: <http://localhost:9080/navigator/?desktop=admin>
7. Click on Plug-ins, and select the Datacap Navigator. Click Edit, then Load. Next click Save and Close. Next, click Close to exit and then log out from the application.
8. Now, log on to your desktop and test the fix. To change the skip blank pages value, go to NScan application. Open the Advanced tab. In the section Scanner Setup, select or clear the Skip Blank Pages check box.

Note: If you want to revert this fix in the Datacap Navigator, complete the steps 3 - 7 using the backed-up copy of DatacapWebPlugin.jar.

Parent topic: [Troubleshooting tips for IBM Datacap](#)

After image rotation, snippet shows image in its original orientation

In Datacap Navigator, when the image is rotated, snippet for the field shows image in its original orientation.

Symptoms

In Datacap Navigator, when the image is rotated, snippet for the field shows image in its original orientation.

Causes

This issue sometimes occurs when a document is ingested upside down in the Datacap Navigator.

Resolving the problem

User response: If a document is ingested upside down and the user notices it during the Verify process, user can resolve this issue by performing the following steps in the Datacap Navigator.

1. In Datacap Navigator, open Job Monitor.
2. Click the image in the Batch Structure to open the image in the viewer.
3. Use the viewer Actions to rotate the image to the desired orientation.
4. Click Hold button. The image is saved in the new orientation.
5. Repeat the steps 2-4 for other images, if required.

The image snippet or the selected words in the image are now shown in the correct orientation, and you can update the necessary content in the Field Details section.

Parent topic: [Troubleshooting tips for IBM Datacap](#)

AppScan issues: Physical path disclosure and hidden directory detection

When a user runs Taskmaster Web (tmWeb .Net) from the same machine it is installed on, in case of error 403 or error 404, the complete error page path or hidden directories are displayed .

Symptoms

Example: Consider that Taskmaster Web (tmWeb.Net) is installed on machine 1. It has a web.config file, which contains “mode” attribute in <customErrors> tag. This property can have three values:

- *off*
- *on*
- *remoteonly*

Now consider that user A is accessing Taskmaster Web from a remote machine, and user B is accessing the Taskmaster Web from same machine on which it is hosted, that is, machine 1. While navigating the Taskmaster Web, a page experiences error 403 or error 404.

If the mode property is set to **remoteonly**:

- The remote user A will see the a user-defined error message, without any physical path disclosure.
- However, the user B will be redirected to a technical error page that displays the complete physical path.

If the mode property is set to **on**:

- In this case, both users A and B will be re-directed to an error page that shows a user-defined error message, without any physical path disclosure.

If the mode property is set to **off**:

- In this case, both users A and B will be redirected to the default technical error page of ASP.NET, with physical path disclosure.

Resolving the problem

User response:

Important: In order to avoid displaying the technical error page with physical path and hidden directories, as a best practice it is recommended that no user should run Taskmaster Web (tmWeb Net) on the same machine where it is installed.

To fix this issue, complete the following steps:

1. On the machine where Taskmaster Web (tmWeb .Net) is installed, open the web.config file, located in the directory C:\Datacap\tmweb.net.
2. **To fix Error Page Path Disclosure:**
 - a. Under the <system.webServer> node, add the code-snippet highlighted below:

```
<system.webServer>
....
.....
  <httpErrors errorMode="Custom">
    <remove statusCode="404" />
    <error statusCode="404" path="/error.aspx"
responseMode="ExecuteURL" />
  </httpErrors> </system.webServer>
```

- b. Under the <system.web> node, add the code-snippet highlighted below:

```
<system.web>
....
```

```
.....  
<customErrors mode="On" defaultRedirect="~/error.aspx"/>  
</system.web>
```

3. To fix Hidden Directory Detection:

a. Under the `<system.webServer>` node, add the code-snippet highlighted below:

```
<system.webServer>  
  <httpErrors errorMode="Custom">  
    <error statusCode="403" subStatusCode="14" path="/error.aspx"  
    responseMode="ExecuteURL" />  
  </httpErrors>  
</system.webServer>
```

4. After updating the web.config file, it is recommended to restart the Internet Information Server (IIS) web server.

Parent topic: [Troubleshooting tips for IBM Datacap](#)

Document integrity check fails to identify batch level variables “MIN TYPES” and “MAX TYPES”

In the document hierarchy, at the batch level, document integrity check fails to identify batch level variables *MIN TYPES* and *MAX TYPES*.

Symptoms

User sets either one or both the variables *MIN TYPES* and *MAX TYPES* at batch level node in document hierarchy. However, the document integrity check passes for a batch that does not contain documents more than the value of *MIN TYPES* and less than the value of *MAX TYPES*.

Causes

In setup DCO there are variables named *MIN_TYPES* and *MAX_TYPES* at every level in hierarchy. However, at the batch level, the variable names do not contain underscore and have spaces instead, such as "*MIN TYPES*". The document integrity code however checks for *MIN_TYPES* and not for *MIN TYPES*

Resolving the problem

Instead of setting variables *MIN TYPES* and *MAX TYPES*, add new variables *MIN_TYPES* and *MAX_TYPES* and set the desired value to these variables.

Parent topic: [Troubleshooting tips for IBM Datacap](#)

Troubleshooting Datacap security and authentication

You might encounter problems when you start the Datacap system or log on to a component. For example, in a client/server environment, you might see a problem with the automatic encryption key import feature.

- [Automatic key import feature fails to apply the new encryption keys](#)
In a client/server environment, the latest encryption keys must be applied when you start a Datacap component. If encryption keys are not applied, you receive authentication errors.

Automatic key import feature fails to apply the new encryption keys

In a client/server environment, the latest encryption keys must be applied when you start a Datacap component. If encryption keys are not applied, you receive authentication errors.

Symptoms

In a client/server environment, when an Datacap component is started on a client computer for the first time, the component cannot communicate with the Datacap server. Authentication errors result.

Causes

This problem can occur when the automatic key import feature fails to apply the new encryption keys as expected when you start the component. This issue impedes communication between the client computer and the Datacap server.

Resolving the problem

Ensure that you imported the latest keys from the Datacap server keystore by using the documented procedure in *Importing encryption keys to IBM® Datacap computers*. Restart the Datacap component. If communication problems still persist, import the keys manually.

To import the keys manually, copy the `dcskey.exe` file from the Datacap server to the client computer. Then, run the `dcskey.exe /i` command from the Microsoft Windows desktop, either from the Start > Run command or from a command prompt.

Parent topic: [Troubleshooting Datacap security and authentication](#)

Related information

[Importing encryption keys to Datacap computers](#)

Troubleshooting Rulerunner

You might need to troubleshoot problems when you use Rulerunner to automatically process background tasks when batches are pending.

You run these steps to troubleshoot Rulerunner.

- [First things to check](#)
When you troubleshoot the Rulerunner Service, you can complete some tasks to ensure that your system is set up correctly. For example, ensure the correct version of Datacap is installed on all computers. The tasks also include running a batch manually, and checking the Windows Event View logs on the Rulerunner server.
- [Viewing Windows Event Viewer logs on Rulerunner server](#)
You can open the Windows Event Viewer and view the Application and System Logs on the Rulerunner server that hosts the Rulerunner Service. The Rulerunner server must be running on Windows.
- [Stopping the Rulerunner Service](#)
You must stop the Rulerunner Service to do system maintenance and back up your environment. Also, to upgrade or remove Datacap software, or clear the Rulerunner Service service memory.
- [Enabling Rulerunner processing logs](#)
Rulerunner processing logs can be created on the Rulerunner server that hosts Rulerunner. Task-specific logs can be created on the Server that hosts Datacap and the application files.

- [Starting the Rulerunner Service](#)
After you set up or change the Rulerunner Service, go to the Rulerunner Manager and start the Rulerunner service.
- [Viewing Rulerunner processing logs](#)
The Rulerunner processing logs provide information to help identify the cause of error conditions. The error conditions might include Datacap server connection failures, inconsistent domain account group names for authentication, or incorrect database path structures in the Application Manager.

First things to check

When you troubleshoot the Rulerunner Service, you can complete some tasks to ensure that your system is set up correctly. For example, ensure the correct version of Datacap is installed on all computers. The tasks also include running a batch manually, and checking the Windows Event View logs on the Rulerunner server.

Make sure that Datacap 9.0 or later is installed on all of the computers in your system.

Ensure that you can run a batch manually by using a Datacap client on a workstation or from a browser. If you cannot manually run batches through the tasks that Rulerunner is to process, your application setup needs to be corrected. Review your installation and configuration of the application and review the installation and configuration topics to troubleshoot your setup.

If you can run a batch manually, through the tasks that Rulerunner is configured to process. And your application is set up properly. And you need to do Rulerunner specific troubleshooting. Follow the instructions in [Viewing Windows Event Viewer logs on Rulerunner server](#) to stop Rulerunner, enable logging, restart Rulerunner, and check the log files for errors.

If Rulerunner starts then stops immediately and the log files indicate that Rulerunner cannot connect to the Datacap Server Service:

- If a firewall is enabled on the Server hosting the Datacap Server Service, ensure that TCP/IP port 2402 was opened.
- When you use Windows Authentication, check with your system administrator to make sure that you selected the correct Windows Authentication template in the Datacap Server Manager.

Parent topic: [Troubleshooting Rulerunner](#)

Viewing Windows Event Viewer logs on Rulerunner server

You can open the Windows Event Viewer and view the Application and System Logs on the Rulerunner server that hosts the Rulerunner Service. The Rulerunner server must be running on Windows.

Procedure

To view Windows Event Viewer logs on Rulerunner server:

1. On the Rulerunner server, in the Start menu click Control Panel > Administrative Tools > Event Viewer > Windows Logs > Application.
2. Select Action > Refresh to ensure that you are seeing all recent messages.
3. Locate the first or last event from the most recent Rulerunner Service run and double-click the entry to view the properties of the event.

Ignore the message:

```
The description for Event ID ( 0 ) in Source... cannot be found...  
The following information is part of the event:
```

The information that follows this text is relevant.

The first event reads:

```
Service Started
```

Subsequent events indicate that the individual threads have each started. There are three initialization attempts with intervals of 2, 4 and 8 seconds apart and errors are logged if they occur.

Any warning or error events indicate a problem, the first warning, or error is typically the most significant.

4. Select System. Entries for individual events are displayed in the details pane.
5. Double-click an entry to view the properties of the event. Messages indicate that the Rulerunner Service was able to log on.

Any warning or error events indicate a problem, the first warning, or error is typically the most significant.

Parent topic: [Troubleshooting Rulerunner](#)

Stopping the Rulerunner Service

You must stop the Rulerunner Service to do system maintenance and back up your environment. Also, to upgrade or remove Datacap software, or clear the Rulerunner Service service memory.

About this task

If you try to stop the Rulerunner Service service while it is processing job tasks, the service completes its current work before it shuts down.

Procedure

To stop the Rulerunner Service:

1. On the Rulerunner server, in the Start menu select IBM Datacap Services > Datacap Rulerunner Manager. If the User Account Control window opens, click Yes.
2. If the Status is Stopped, close the Rulerunner Manager window.
3. If the Status is Running, click Stop.

Parent topic: [Troubleshooting Rulerunner](#)

Enabling Rulerunner processing logs

Rulerunner processing logs can be created on the Rulerunner server that hosts Rulerunner. Task-specific logs can be created on the Server that hosts Datacap and the application files.

About this task

You must have 10 GB or more of disk space on each server for storing these logs. Remember to change the level of detail that is written to the logs or to turn them off when you no longer need them.

Procedure

To enable Rulerunner processing logs:

1. On Rulerunner server, in the Start menu click IBM Datacap Services > Rulerunner Manager.
2. If the Status displayed for the Rulerunner Service is Running, click Stop.
3. Click the Rulerunner Login tab.
4. Select the Windows Authentication option, and click Connect.
5. Click the Logging tab.
6. On the Rulerunner Log tab, move the Application Event Log level slider to Serious and critical.
7. On the Rulerunner Log tab, move the Thread Log level slider to All, then click Save.
8. Ensure that the Output to folder option is selected. Change or make a note of the path in the Output to field. This path is where the logs are created and you need this information later.
9. Change other options as wanted, then click Save.
10. Click the RSS Log tab, move the sliders to All for Level of detail written to the RRS logs slider and the Severity level of messages logged.
11. Select the Batch Log option. Change other options as wanted, then click Save.
12. Click the Rulerunner Login tab.
13. Click Disconnect.
Important: Complete the following steps to ensure that the Rulerunner authentication credentials are set properly.
14. When you are using:
 - o Windows Authentication - Select the Windows Authentication option.
 - o Datacap Authentication - Select the Datacap Authentication option, enter the user ID and Password that is used by Rulerunner, the name of the Rulerunner server as the Station ID.
15. Click Save.
16. Close the Rulerunner Manager window.

Parent topic: [Troubleshooting Rulerunner](#)

Starting the Rulerunner Service

After you set up or change the Rulerunner Service, go to the Rulerunner Manager and start the Rulerunner service.

Procedure

Follow this procedure to start the Rulerunner Service.

1. Go to Start > IBM Datacap Services > Datacap Rulerunner Manager. If the User Account Control window opens, click Yes.
2. If the Status is Stopped, click Start. The Status changes to Running.
3. Close the Rulerunner Manager window.

Parent topic: [Troubleshooting Rulerunner](#)

Viewing Rulerunner processing logs

The Rulerunner processing logs provide information to help identify the cause of error conditions. The error conditions might include Datacap server connection failures, inconsistent domain account group names for authentication, or incorrect database path structures in the Application Manager.

About this task

When you enable logging and restart Rulerunner, Rulerunner processes batches and writes the processing information to the logs. You can view the various Rulerunner logs to get this information.

Procedure

To view Rulerunner processing logs:

1. On the Rulerunner server, start Windows Explorer, open the folder in which the logs were created. The default is C:\Datacap\. The folder is defined by Rulerunner Manager. Click the Logging - Rulerunner Log tab and select Output to folder.
2. Open the Rulerunner.log file in a text editor such as Notepad and look for errors.
3. Open the Rulerunner thread log files in a text editor. Look for the messages that begin with EstablishConnection. A successful log would be similar to this example:

```
EstablishConnection: Connecting to application [APT].
EstablishConnection: GetServer returned [127.0.0.1].
EstablishConnection: Connecting to TM Server [127.0.0.1:2402].
EstablishConnection: Connecting to Admin DB
[PROVIDER=MSACCESS;DSN=C:\Datacap\APT\APTAdm.mdb;].
EstablishConnection: Connecting to Engine DB
[PROVIDER=MSACCESS;DSN=C:\Datacap\APT\APTEng.mdb;].
EstablishConnection: TM Login [admin station=1].
EstablishConnection: Login successful.
```

4. Error conditions can include one or more of the following examples:
 - If you are connecting to the Datacap Server fails, then the Datacap Server Service is not responsive or a network failure occurred.
 - If you are connecting to the Admin or Engine database of the application fails, use the Datacap Application Manager. To ensure that the paths to the databases are correct.
 - If Rulerunner fails to log in to the Datacap Server, do the following steps.
 - Ensure that the Rulerunner Service is set up properly by reviewing the Rulerunner related instructions.
 - Ensure that the Rulerunner domain account was added to the correct security group. And, in your Datacap application, a matching group is set up with a name that ends in the complete domain name (for example, XYZ.com, not XYZ).
5. On the Server, start Windows Explorer, go to and open the batches folder of the application. The default folder is C:\Datacap\Application\Batches.
6. Open the taskname_rrs.log in a text editor and look for errors. For example, open pageid_rrs.log or recognize_rrs.log.

During processing, when Rulerunner fails to run a batch from a particular application, there is a delay interval before the next attempt. Rulerunner doubles the delay each time 2 - 64 seconds.

Parent topic: [Troubleshooting Rulerunner](#)

Troubleshooting FastDoc

You might need to troubleshoot problems when you use FastDoc to scan, process, verify, and export documents.

Some of the common issues that might require troubleshooting include these issues:

- FastDoc client issues like poor barcode recognition, documents types are not assigned, and scanners are not displayed.
- FastDoc application issues such as extra data is found in captured data and index data is not picked up in a zoned field.
- FastDoc client export issues that include errors during the export and SharePoint Connector upload errors from the Export_rrs.log file.

- [Barcode recognition is poor](#)
FastDoc is set up to capture the value of any type of barcode. If you cannot capture barcode values, the barcodes might be blurred or somehow damaged.
- [Document type is not automatically assigned](#)
FastDoc does not automatically assign a document type to a document that was set up by your Administrator. There is something wrong with the fingerprint of that document.
- [Scanner is not listed as an option](#)
When you try to select a scanner from the TWAIN Scan Source list, your scanner is not listed as an available option.
- [Extra data included in captured data](#)
A zone for a field on a particular document is created incorrectly, such as with the wrong location or the wrong size. Unwanted data can be included in the field during document processing. For example, the field label or data from the previous line or the next line might be included in the field.
- [Index data not picked up automatically in zoned field](#)
A zone for a field on a particular document is created incorrectly, such as in the wrong location or the wrong size. The index data for the field is not automatically picked up.
- [Correcting export errors](#)
An error message displays when you export images from FastDoc.
- [SharePoint upload returns file check-in error in Export_rrs.log](#)
The required columns in a SharePoint library do not include default values. FastDoc can upload the first document in the batch and update the properties for that document. But FastDoc cannot complete the SharePoint check-in process for the first document.
- [SharePoint upload returns a list not found error in Export_rrs.log](#)
The SharePoint Administrator who created a library, then later changed the name that is displayed to users. You used the display name when you set up your export to SharePoint Document Type Settings, the upload to SharePoint fails.
- [SharePoint upload returns a properties update error in Export_rrs.log](#)
The characteristics of the index values passed by FastDoc to SharePoint must match the characteristics of the SharePoint columns.
- [FastDoc labels cannot be displayed](#)
Labels are not displayed in "Configure documents, pages, and fields" icon, due to Windows environment issue.

Barcode recognition is poor

FastDoc is set up to capture the value of any type of barcode. If you cannot capture barcode values, the barcodes might be blurred or somehow damaged.

When barcode recognition is poor, you can use the Resolution dial on the TWAIN Scan or ISIS Scan panel to increase the resolution before you begin scanning the documents. Then, during the Verify process, you redraw the zone around the barcode.

Attention: When you scan an image with a different DPI, you must create a new fingerprint. If the barcode type is set to Any, change it to the barcode type that is used.

You can also reprint the barcodes and apply them to the problem pages or enter the barcode values manually.

Parent topic: [Troubleshooting FastDoc](#)

Document type is not automatically assigned

FastDoc does not automatically assign a document type to a document that was set up by your Administrator. There is something wrong with the fingerprint of that document.

When you click Fingerprint, the fingerprint ID number is displayed with Add and Delete options. If the fingerprint is new, zone the fields to save the positions to the new fingerprint. The new fingerprint is recognized the next time that you run the task. If the wrong type is matched, click Fingerprint and click New to force the creation of a new fingerprint. Optionally, you can click Delete to remove the fingerprint that was matched if it is no longer used.

Parent topic: [Troubleshooting FastDoc](#)

Scanner is not listed as an option

When you try to select a scanner from the TWAIN Scan Source list, your scanner is not listed as an available option.

Use the scanner manufacturer instructions and driver software to ensure that your scanner and scanner driver are installed correctly and that you can use the scanner to scan documents successfully outside of FastDoc.

Parent topic: [Troubleshooting FastDoc](#)

Extra data included in captured data

A zone for a field on a particular document is created incorrectly, such as with the wrong location or the wrong size. Unwanted data can be included in the field during document processing. For example, the field label or data from the previous line or the next line might be included in the field.

To correct this problem, click Fingerprint to see the Fingerprint ID of the document. On the Fingerprint dialog, click Delete to remove the fingerprint. Then, rescan the document, reassign the Document Type to the document, and re-create the zones so that unwanted data is not captured.

Parent topic: [Troubleshooting FastDoc](#)

Index data not picked up automatically in zoned field

A zone for a field on a particular document is created incorrectly, such as in the wrong location or the wrong size. The index data for the field is not automatically picked up.

To correct this problem, click Fingerprint to see the Fingerprint ID of the document. On the Fingerprint dialog, click Delete to remove the fingerprint. Then, rescan the document, reassign the Document Type to the document, and re-create the zones so that the index data is correctly included.

Parent topic: [Troubleshooting FastDoc](#)

Correcting export errors

An error message displays when you export images from FastDoc.

About this task

The following error message displays during export:

```
There was a problem exporting batchXYZ.  
Some documents may not have been exported.  
Check your settings.
```

You must ensure that your export settings are set up correctly and look in the Export_rrs.log file of the batch to find any problems.

Procedure

To correct export settings:

1. Open FastDoc as Administrator, scan a sample of the Document Type that you are having problems with, and click Next Task.
2. After Recognize completes and the Verify panel is displayed with the image displayed in the Active Image pane, click Document Type Settings and ensure that your export settings are correctly set up.
3. Open the batch folder and the Export_rrs.log file, and search for lines that contain the words error or abort. The surrounding lines contain information about the error condition.
4. When you export to SharePoint, see the following topics when the error is related to uploading to SharePoint:
 - [SharePoint upload results in Could not check in file error in export_rrs.log](#)
 - [SharePoint upload results in Could not find list error in export_rrs.log](#)
 - [SharePoint upload results in Could not update properties error in export_rrs.log](#)
5. When you export to IBM® FileNet® Content Manager, verify the following export settings, make corrections where needed, and rerun the export.
 - The URL of the Content Manager into which you want to export documents
 - The Object Store and Document Class on the Content Manager
 - The folder on the Content Manager where you want to store the documents

Parent topic: [Troubleshooting FastDoc](#)

SharePoint upload returns file check-in error in Export_rrs.log

The required columns in a SharePoint library do not include default values. FastDoc can upload the first document in the batch and update the properties for that document. But FastDoc cannot complete the SharePoint check-in process for the first document.

This problem results in a checked-out SharePoint document that only the FastDoc user can see. Other SharePoint users cannot see the document.

In addition, FastDoc is unable to continue processing the remaining documents from the same batch. Those documents were not uploaded to SharePoint at all.

When you get this error, ask your SharePoint Administrator to edit the Document Library Settings for each required column. Add a default value for each, for example, No Value Available. Then, delete the document that was uploaded but not checked in.

After the SharePoint Administrator changes Document Library Settings for each required column, reprocess the entire batch of documents by using the FastDoc Client.

Parent topic: [Troubleshooting FastDoc](#)

SharePoint upload returns a list not found error in Export_rrs.log

The SharePoint Administrator who created a library, then later changed the name that is displayed to users. You used the display name when you set up your export to SharePoint Document Type Settings, the upload to SharePoint fails.

The SharePoint site URL must reflect the original name, not the name that is displayed to users. Obtain the original name from your SharePoint Administrator and change the URL in FastDoc.

Parent topic: [Troubleshooting FastDoc](#)

SharePoint upload returns a properties update error in Export_rrs.log

The characteristics of the index values passed by FastDoc to SharePoint must match the characteristics of the SharePoint columns.

If the index values are not set up properly in FastDoc, they are rejected by SharePoint. The `Could not update Properties` error can occur for a number of reasons, and the error message provides information for the error. For example, if you passed an index value to a SharePoint Date column without identifying it as a Date in FastDoc.

```
Error code: '0x8102001c' Invalid date/time value
A date/time field contains invalid data.
Please check the value and try again.
```

Parent topic: [Troubleshooting FastDoc](#)

FastDoc labels cannot be displayed

Labels are not displayed in "Configure documents, pages, and fields" icon, due to Windows environment issue.

In FastDoc Admin, under the "Configure documents, pages, and fields" icon, the labels are not displayed. The labels that are related to the validate fields option for a field property do not appear.

This is an environment issue that is specific to Windows 2008 R2. To overcome this problem, you can use a version of windows that is compatible with Fastdoc (example, Windows 7, Windows 10).

Parent topic: [Troubleshooting FastDoc](#)

Troubleshooting Datacap web services

If the file upload fails on large files or the PUT request method endpoints are not working, you can check to ensure that your configurations are correct.

Slow log on times possibly caused by network related configurations.

It may be possible to improve log on times by adding the IP address of the Datacap Server to the hosts file on the web services computer.

The file upload fails for SetFile or UploadFile

If the file upload fails on large files when you are using the POST methods SetFile or UploadFile, check that `maxAllowedContentLength` and `maxRequestLength` are configured correctly. You can adjust the value of the `maxAllowedContentLength` and `maxRequestLength` settings in the web.config file. The maximum size message that can be uploaded depends on your web server, proxy server, and client.

The PUT request method GrabBatch, ReleaseBatch, or SetPageFileName does not work

If the PUT request method GrabBatch, ReleaseBatch, or SetPageFileName is not working, ensure that the WebDAV Publishing role service is not installed and that the PUT request method is allowed. The WebDAV Publishing role service prevents the Datacap Web Services PUT method from functioning. You can verify in the Server Manager that the WebDAV Publishing role service is not installed.

- In the Start menu on the web server, select IBM Datacap Services > Datacap Server Manager.
- In the Server Manager hierarchy pane, expand Roles and select Web Server (IIS).
- In the Web Server (IIS) pane, expand Role Services. Under Common HTTP Features, ensure that the WebDAV Publishing role service is not installed.

Related information:

[Verifying that IIS components are installed](#)
[Datacap Web Services REST API methods](#)

Log files

To obtain adequate logging information for troubleshooting purposes, you must enable logging for the client and for the Rulerunner Service.

Adequate logging includes information about a client, such as Datacap Desktop and the Rulerunner Service, which applies rules to an action that a task requires. Maximum logging can be enabled all of the time for a test or development system. However, because maximum logging negatively impacts system performance, you can set the logging to the minimum level on a production server.

- [Enabling logging for Datacap Desktop](#)
To enable logging for Datacap Desktop, you must edit the dcDesktop.exe.config file and enable logging for Rulerunner Service. To confirm that logging is functional, complete an action in Datacap Desktop and review the Rulerunner Service RRS.log file and the dcdesktop.log file.
- [Enabling logging for FastDoc](#)
To enable logging for FastDoc, you must edit the FastDoc.exe.config file and enable logging for Rulerunner Service. To confirm that logging is functional, complete an action in FastDoc and review the RRS.log file and the FastDoc.log file.
- [Enabling logging for Rulerunner Service](#)
To enable logging for Rulerunner Service, you must configure the logging options in Rulerunner Manager. To confirm that logging is functional, complete a task where rules are applied, and review the rulerunner.log file, therulerunner_thread_atm.log file, and the *task_rrs*.log file.
- [Enabling logging for Datacap Server service](#)
To enable logging for Datacap Server service, you must configure the logging options in the Datacap Server Manager. To confirm that logging is functional, complete a task in a workflow and review the tms.log file.
- [Enabling logging for the Datacap Web Client](#)
To enable logging for Datacap Web Client, you must edit the server.ini file. To confirm that logging is functional, complete a task in Datacap Web Client and review the tmweb.log file and the pacuerr.log file.
- [Enabling the Datacap Web Services log](#)
When you enable logging for Datacap Web Services in the c:\Datacap\wtm\web.config file, the wTM and aTM logs are created.

Related information:

[Set Rulerunner logging by application and task](#)

Enabling logging for Datacap Desktop

To enable logging for Datacap Desktop, you must edit the `dcDesktop.exe.config` file and enable logging for Rulerunner Service. To confirm that logging is functional, complete an action in Datacap Desktop and review the Rulerunner Service `RRS.log` file and the `dcdesktop.log` file.

About this task

Although there are different levels of logging, these steps enable maximum logging to troubleshoot Datacap Desktop when problems occur.

Procedure

To enable logging for Datacap Desktop:

1. Enable executable file logging.
 - a. Go to the `\Datacap\DcDesktop` folder and open the `dcDesktop.exe.config` file.
 - b. In the User Settings section, enter these values.
 - `WriteLog = True`
 - `LogSeverity = 5`
 - `LogPath = dcdesktop.log`
 - `LogOverwrite = True`
 - `LogFlushBuffer = True`
 - `LogShowTime = True`
 - c. Save the file.
2. Enable Rulerunner Service logging.
 - a. Start Datacap Web Client and log in to the application that is to be run, and select the Administrator tab.
 - b. Expand the Workflow, highlight the task for which logging is needed, and click Setup.
 - c. In the Rulerunner Service log settings section, under Rulerunner settings, set the Rulerunner service log field to 5 and select the Flush Buffer check box.

Note: Selecting the Flush Buffer option slows down the process because every line that is logged is written to the log file on disk, instead of being buffered in memory.
 - d. Scroll to the end of Webpage Dialog and click Save.
3. Start Datacap Desktop and start a task for which logging information is needed. Two log files are generated:
 - o The `X_rrs.log`, where `X` is the name of the task. The file is in the batch folder of the application (for example, `C:\datacap\TravelDocs\batches\20130924.00001\pageid_rrs.log`), and contains information about actions that are completed by the Rulerunner engine.
 - o The `dcdesktop.log.Y` file, where `Y` is the log sequence number. This file is in the `\Users\username\AppData\Local\IBM\IBM Datacap\9.0.0.0` folder, and contains information about the Datacap Desktop executable file.

What to do next

Disable Datacap Desktop Rulerunner logging by setting `WriteLog = False` in the `dcDesktop.exe.config` file. You cannot completely disable the Datacap Desktop executable file logging, but you can create minimum logging by setting Rulerunner service log, Batch log, and Action log level to 0. Be sure to clear the Flush Buffer check box to avoid reduced performance.

Parent topic: [Log files](#)

Enabling logging for FastDoc

To enable logging for FastDoc, you must edit the FastDoc.exe.config file and enable logging for Rulerunner Service. To confirm that logging is functional, complete an action in FastDoc and review the RRS.log file and the FastDoc.log file.

About this task

Although there are different levels of logging, these steps enable maximum logging to troubleshoot FastDoc when problems occur.

Procedure

To enable logging for FastDoc:

1. Enable executable file logging.
 - a. Go to the \Datacap\FastDoc folder and open the FastDoc.exe.config file.
 - b. In the User Settings section, set `Log = True` and save the file.
2. If you are running FastDoc in Local Mode, enable logging for the Rulerunner Service by completing these steps.
 - a. Go to the \Datacap\FastDoc folder and open the BatchProfiles.xml file.
 - b. Locate the `rrslog` tag that is specific to the task for which logging is needed and set `ServiceLog="5"`.
 - c. Save the BatchProfiles.xml file.
3. If you are running FastDoc in Datacap Mode, enable logging for the Rulerunner Service by completing these steps.
 - a. In the Rulerunner Service log settings for the workflow task in your web client administration panel, set the Rulerunner service log field to 5 and select the Flush Buffer check box.
Note: Selecting the Flush Buffer option slows down the process because every line that is logged is written to the log file on disk, instead of being buffered in memory.
4. Start FastDoc and start a task for which logging information is needed. Two log files are generated:
 - o The `X_rrs.log`, where `X` is the name of the task. The file is in the batch folder of the application (for example, `C:\datacap\TravelDocs\batches\20130924.00001\pageid_rrs.log`), and contains information about actions that are completed by the Rulerunner engine.
 - o The `FastDoc.log`. This file is in the `\Users\username\AppData\Local\IBM\IBM Datacap\9.0.0.xx` folder, and contains information about the FastDoc executable file.

What to do next

Disable the FastDoc executable file logging by setting `Log = False` in the FastDoc.exe.config file. You cannot completely disable the FastDoc Rulerunner logging, but you can create minimum logging by setting Rulerunner service log, Batch log, and Action log level to 0. Be sure to clear the Flush Buffer check box to avoid reduced performance.

Parent topic: [Log files](#)

Enabling logging for Rulerunner Service

To enable logging for Rulerunner Service, you must configure the logging options in Rulerunner Manager. To confirm that logging is functional, complete a task where rules are applied, and review the rulerunner.log file, therulerunner_thread_atm.log file, and the `task_rrs.log` file.

About this task

Although there are different levels of logging, these steps enable maximum logging to troubleshoot Rulerunner Service when problems occur.

Procedure

To enable logging for Rulerunner Service:

1. Start Rulerunner Manager.
2. Click the Rulerunner Login tab.
3. Enter the login information, as needed, and click Connect.
4. Click the Logging tab, and then select Quick Log at the end of the page.
5. Move the slider to Debug.
6. Click Save.
7. On the Rulerunner Login tab, click Disconnect, and exit from Rulerunner Manager.
8. Complete a task for which logging information is needed. These log files are generated:
 - o The RulerunnerX.log, where X is the thread. The default location of the file is C:\datacap, and contains general logging information for the Rulerunner process.
 - o The Rulerunner_thread_X_atmY.log, where X is the name of the task and Y is the log sequence number. The default location of the file is C:\datacap, and contains detailed logging information for the Rulerunner process.
 - o The Z_rrs.log, where Z is the name of the task. The file is in the batch folder of the application (for example, C:\datacap\TravelDocs\batches\20130924.00001\pageid_rrs.log), and contains information about actions that are completed by the Rulerunner engine.

Tip: You do not need to restart the Rulerunner Service after you save the logging configuration settings. The service automatically retrieves the changes after all of the pending tasks are completed. However, you can restart the service to immediately implement the changes.

The Rulerunner Service uses managed logging for all of the log files except the rrs log file. When the log files are full, Rulerunner deletes the oldest file and creates a new file. Managed logging provides the option of keeping logging enabled without using excessive hard disk space.

Parent topic: [Log files](#)

Enabling logging for Datacap Server service

To enable logging for Datacap Server service, you must configure the logging options in the Datacap Server Manager. To confirm that logging is functional, complete a task in a workflow and review the tms.log file.

About this task

Although there are different levels of logging, these steps enable maximum logging to troubleshoot Datacap Server service when problems occur.

Procedure

To enable logging for Datacap Server service:

1. Start Datacap Server Manager.
2. Click the Logging tab, and then click the Datacap log tab.
3. Move the slider to All and select the Output to file check box.
4. Click Save and exit from Datacap Server Manager.
5. Complete a task in a workflow for which logging information is needed. Datacap Server Manager generates 1 - 5 log files with the name tms.log.X.log, where X is the log number. The location of the log

files is in the Datacap installation directory. (The default is C:\datacap.)

Tip: You do not need to restart the Datacap Server service after you save the logging configuration settings. The service automatically retrieves the changes after a few seconds.

The Datacap Server service uses managed logging. After the log files are full, Datacap deletes the oldest file and creates a new file. Managed logging provides the option of keeping logging enabled without using excessive hard disk space.

Parent topic: [Log files](#)

Enabling logging for the Datacap Web Client

To enable logging for Datacap Web Client, you must edit the server.ini file. To confirm that logging is functional, complete a task in Datacap Web Client and review the tmweb.log file and the pacuerr.log file.

About this task

Although there are different levels of logging, these steps enable maximum logging to troubleshoot Datacap Web Client when problems occur.

Procedure

To enable logging for Datacap Web Client:

1. Go to the \Datacap\tmweb.net folder and open the server.ini file.
2. In the [General] section, set WriteLog = 1.
3. In the [Log] section, enter these values.
 - Path = C:\Datacap\TaskRun\tmweb.log (or other valid path and file name).
 - Overwrite = 1
 - ShowTime = 1
 - FlushBuffer = 1
4. Save the server.ini file.
5. Restart IIS.
6. Start Datacap Web Client, log in to the application for which logging information is required, and complete a task in a workflow. Two log files are generated.
 - tmweb.log.X.log, where X is the log sequence number. The file is in the C:\Datacap\TaskRun folder, or the folder that was specified in the Path configuration setting.
 - Datacap Web Client also contains an error log that is always enabled. If the web page itself encounters an error and times out, Datacap generates a file that is called pacuerr.log, which is in the C:\Datacap\tmweb.net\App_Data folder.

What to do next

Disable Datacap Web Client logging by setting WriteLog = 0 in the server.ini file and restarting IIS.

Parent topic: [Log files](#)

Enabling the Datacap Web Services log

When you enable logging for Datacap Web Services in the c:\Datacap\wtm\web.config file, the wTM and aTM logs are created.

Procedure

Enable logging for Datacap Web Services.

1. Navigate to the c:\Datacap\wtm folder and open the web.config file.
2. In the `<IBM.Datacap.Web.Properties.Settings>` section, set `logEnable` to `True`. Enter the other settings values, including the `logPath` and the `logSeverity` settings. You can control the level of detail that is written to the log by entering one of the following values in the `logSeverity` setting.
 - o 0 - No messages are logged
 - o 1-3 - Logs warnings and exceptions
 - o 4-6 - Logs information, warnings, and exceptions
 - o 7-9 - Logs the maximum amount of detail

Example

```
<setting name="logSeverity" serializeAs="String">
<value>4</value>
</setting>
<setting name="logOverwrite" serializeAs="String">
<value>False</value>
</setting>
<setting name="logPath" serializeAs="String">
<value>c:\</value>
</setting>
<setting name="logTime" serializeAs="String">
<value>True</value>
</setting>
<setting name="logFlushBuffer" serializeAs="String">
<value>True</value>
</setting>
<setting name="logEnable" serializeAs="String">
<value>True</value>
</setting>
```

Parent topic: [Log files](#)

Best Practices for optimal text recognition

This document provides information about text recognition in general, and detailed guidance to achieve the best results when processing documents by using IBM® Datacap.

To know about best practices for optimal text recognition, see [Best Practices for optimal text recognition in IBM Datacap](#).

Best practices for source control of Datacap applications

This document discusses the implications of using source control on a Datacap application.

Datacap applications are consisted of a collection of files and databases. Source control can help track when an application has been updated and the user that performed the update. This document discusses the implications of using source control on a Datacap application.

- [Parts of a Datacap Application](#)
A Datacap is primarily a collection of files. Here is a review of the components and how they relate to source control.
- [Source Control of Datacap applications](#)
There are several components that together comprise a Datacap application. The database portion is the

primary piece that does not fit into complete source control solution because it is not a "file" unless Microsoft Access is being used.

Parts of a Datacap Application

A Datacap is primarily a collection of files. Here is a review of the components and how they relate to source control.

Batch Directories

By default, a batch directory is in a subdirectory of the main application. The batch directory can be moved to any location and is not required to be under the application. Typically there is no reason to source control a batch. It is created and then usually not kept as a whole unit. What is saved from a batch varies from application to application and usually ends up in an external repository. Batches are never versioned. A batch is created and then lives in its final state until important artifacts are saved and discarded. A batch is not expected to be managed by using a source control system.

A batch directory can also exist outside of the application by using the web service to run batches. In this environment, batches are created in its own protected location and are deleted after completion.

Databases

A Datacap database stores several types of data:

- Administrative
- Engine
- Fingerprint
- Lookup
- Other custom data

Administrative Database

Administrative settings regarding users along with some configuration settings such as shortcuts and some task configuration settings.

Engine Database

The Engine database contains the information regarding the batches. The batch IDs, who ran the batch, time for the batch and statistics.

Fingerprint Database

The fingerprint database holds the information regarding the fingerprints that have been created for the application.

Lookup Database

The lookup database is a custom database. APT provides a sample database that could be used as-is, but can also have its schema altered by the user.

Custom Database

Datacap actions support generic SQL.

Database Details

The following table identifies the following for each type of database:

- If a specific schema defined by the Datacap application is required.
- If the database is supported by the Application Copy tool to migrate database data. The tool does not update database schemas. If the database schema is not a Datacap schema, then the user is responsible for creating necessary scripts and steps to move the data from the test environment to a production environment.
- If the database data is typically moved from test to production.

Table 1. DB Types

DB Type	Datacap Schema	Application Copy	Production Rollout
Administrative	Yes	Yes	Changes need to be moved to production. If there are differences in production vs. test, such as different users, then a partial move might be performed.
Engine	Yes	Yes	Typically changes are not moved to production. An updated schema should be applied to the production database, but the table data is not moved to production.
Fingerprint	Yes	Yes	Changes need to be moved to production.
Lookup	No	No	Depends on application. Typically it would be expected to be moved.
Custom	No	No	Depends on application. Typically it would be expected to be moved.

Database Source Control Implications

For a test or demonstration application, an Access database may be used. An Access database is a single file that would be treated as a binary “blob” object by source control. Typical test and production systems use a production grade database such as DB2.

The changes to the database would be controlled and tracked by the database software and not by a source control system.

A rollout from a test environment would need to be coordinated so that new files are placed into production at the same time as any database updates.

Fingerprint Directory & Fingerprint Service

The fingerprint directory contains the images, CCO files, and FPXML files that are used for fingerprinting. The images and CCO files are binary while the FPXML is text. This directory typically lives under the application’s root directory. If the fingerprint web service is used, then the location of the fingerprints could be elsewhere in a network environment.

A fingerprint may need to be moved from test to production depending on the application. For example, with a learning application, such as APT, fingerprints are created during production. There would not be fingerprints to move from test to production. Similarly, there would not be fingerprints to manage in source control.

When using a form application, fingerprints are typically pre-created in a development environment. These fingerprints would ultimately be moved to a production environment and would likely want to be source controlled.

As mentioned, fingerprints may be managed by a fingerprint service and potentially could be shared with multiple applications. Moving these fingerprints to a production environment needs to be synchronized with moving of the application updates.

Fingerprints maintained by the fingerprint service would be considered a separate component to manage with source control as it is not necessarily tied to an application or a single application.

DCO Directory

The DCO directory has a `_DCO` suffix. It contains the setup DCO object and a number of configuration files, such as task settings files, INI files, key files, and any other custom files that were created by the user for their application. This directory can be a mix of text and binary files.

The files in the DCO directory may not be known by Datacap Studio or FastDoc. For example, the user could create one or more custom keyfile that is used by a locate action. Other custom files could be accessed by custom actions.

The rules directory is a child of this directory. The rules directory contains all of the rules for the application. This directory will also contain the binary files of compiled rulesets that are added to the application along with their configuration settings files.

For some applications, such as APT, this directory contains additional files such as “Add Vendor Demo” and “Fingerprint Maintenance Tool” along with corresponding subdirectories that contain translated resources for these applications. These files could be source controlled although they are typically only updated with a new product install and are not considered part of the “application” as they are application utilities.

Additional Custom Directories

An application could have any number other additional directories in any given location. They could be somewhere under the application tree or somewhere else on the machine or network. One example would be the “Graphics” directory that is part of the APT application. An application that performs signature matching can have one or more directories to store reference images. Any kind of custom directory could be created and accessed by standard actions or custom actions.

These directories are not created or known by Datacap Studio or FastDoc. The only way they could be found by these two applications would be by scanning the directories under the application folder, provided they are within the application folder.

Custom Panels

There are two kinds of customizable panels. Datacap Navigator and Datacap Desktop.

Navigator

The customizations to the Navigator files are stored in the application directory in a folder called “Navigator”. This directory can contain additional subdirectories and the files are text-based files that are created by the Navigator client.

Datacap Desktop

It is possible to create custom panels for applications that are displayed from within Datacap Desktop. These panels are written in C# using Microsoft Visual Studio and the binary files are in the Datacap Desktop directory. The source files used to build the panels would be managed separately from the application.

If the panels are updated, then it is a manual process for the user to move them to the production directories as required. They are not copied with the Application Copy tool.

Custom Actions

A user can create custom actions that expand the capabilities of the Datacap application to provide special processing that is not included in the base system. This will be source code written in C# or in Visual Basic Script. For both types of actions, the source code used to build the binaries would be created in Microsoft Visual Studio. It is also possible to use a plain text editor to create VB actions with a plain text editor because it does not need to be pre-compiled.

C# Actions

The binary files for this kind of custom action can live anywhere. There are two different ways to create a C# action. There can be a single DLL or a DLL with a corresponding RRX. If it is a single DLL, it can reside in the RRS directory or in the application rules directory.

If the DLL is a combination of a DLL and RRX file, then the RRX resides in the RRS or application rules directory and the DLL can reside in any directory but typically it is placed in the \Datacap\DCShared\Net directory.

VB Actions

An RRX file that is a visual basic script file needs to sit in either the RRS directory or in the application's rules directory.

Parent topic: [Best practices for source control of Datacap applications](#)

Source Control of Datacap applications

There are several components that together comprise a Datacap application. The database portion is the primary piece that does not fit into complete source control solution because it is not a "file" unless Microsoft Access is being used.

Introduction

Assuming that an application is within source control, it is assumed that there are several reasons for source control.

- Moving changes to production in a controlled manner.
- Controlling changes to an application.
- Tracking who changed an application.

Source Control Application Rollout

When strictly controlling who updates an application with source control, it would be assumed that there would not be a manual step to place the software in the production system. There would need to be an automated process to take what is in source control and place it into the target production system.

This would need to be able to place the contents of the application directories along with any directories that are external to the application. So, if there is a directory that is used externally, possibly fingerprints or other custom directories, the automated process would need to take those directories and files from source control and place them into the target systems.

As the database portion is not managed by source control, some kind of custom step is needed to move any database updates to the production database in a synchronized manner.

Manual Source Control

It is possible to source control an application without direct source control repository integration with Datacap Studio or FastDoc. Typically, source control has a directory structure that is the location of the managed source files that have been retrieved from source control and placed in a managed source area where the developer has access. Source control applications such as RTC and GIT can monitor a directory and all of the files and subdirectories.

A manual solution would have the application sitting in source control with all of the required folders and files. In the case where folders are outside an application, then the developer would need to manually copy them to a subdirectory to manager it together with the application and then the drop process that occurs after check-in would need to know that a directory needs to live in a location outside the application. This is similar to a kind of build script that runs after files are checked into a source control system.

Monitoring the Application Directories

There are two basic ways that a source control system tracks changes in a directory.

In some source control systems, such as RTC and GIT, the managed directory is scanned for changes and the system identifies the files that have been changed compared to the files that exist in the source control system.

Another approach used by some source control systems, such as TFS, requires the user to identify each file that has been updated to allow it to be checked into the source control system.

A user can directly work on an application and make changes as needed. When the user has a consistent set of changes, the user can have the source control scan the directories and check in the changed files. These would then go through the source control's process to distribute changes to production as defined by the user. It is possible that it will placed into some area and can be moved to production at an interval decided by the user.

Text Files and Binary Files

A Datacap application is comprised of text readable files and binary files. Most of the configuration files are stored as XML. These include but are not limited to the setup DCO, collection.xml, rulesets, ini, key, ruleset config, rrx, navigator configuration files, app files, and user created text files.

An application also can have files that are binary such as images, cco, custom actions, compiled rulesets, an Access database, and user created binary files.

Multiple Developers

Datacap Studio has built-in locking of rulesets, setup DCO, and profiles. In a source control environment, this locking is not able to prevent simultaneous updates because developers are working on separate copies of the same application. This implies that all of the coordination of application updates between multiple developers must be handled exclusively by the source control system.

Simultaneous Updates

As one developer updates an application, another developer working on that application can use the source control system to obtain the updated files. If the user accepting the updates has not updated the same file, then they can be added to the application without issue.

If the user accepting the updates has updated one or more of the same files, then the changes must be reconciled. If the changes are to a text-based file, then it can be possible to merge the changes together as long as they do not intersect. It may take manual intervention to reconcile the changes. This is similar to what is done when a code source file is updated. If different areas of the source code are updated, it may be possible to easily integrate changes. If the same area is updated, then manual steps are required to analyze the changes and integrate. In some cases it could mean to accept the updated file as-is and then reapply the changes.

If the file updated by both users is binary, then there is no other alternative but to accept the binary file as is and reapply changes with whatever the tool is that modifies the binary files, if they are still necessary.

Some text files, while are technically user editable, may be too complex to merge by hand. Some examples are the ruleset configuration files. These kinds of files may need to be treated like a binary file when accepting updates and applying changes.

Some repository systems, allow a user to lock a file. Locking a file would prevent another user from checking in a change until the first user has checked-in their own change and unlocked the file. Depending on the desired working style of the developers, this approach could be taken to lock some key files through the source control system when they are going to be updated. This would for other developers to wait for the updates to be finished before making their own updates to a locked file.

Shared Databases

As mentioned, only Access databases have a possibility of being managed as binary files. For any production application, Access database is not used. This means that if multiple developers are working on the same application, the database they are using is likely a shared development database on a separate machine.

If two developers are truly working simultaneously, it is possible for one to make a change to the database that requires the second developer to obtain updates to keep working.

All changes to the database would be controlled and tracked separately by the database itself, not a source control system.

Deploying Database Changes to a Production Environment

When an application has been updated to a state where it is ready to be rolled out to a production environment, the changes made within the source control system need to be moved out synchronized with the database changes.

The traditional way of moving an application from test to production is using the Application Copy tool. This can be run in an interactive GUI or command line. It is possible to have the source system run this tool every time there is a check-in and move the files and database changes to the production system.

It is likely that it is not practical to move out to production with every check-in, so there could be a manual step of some sort that runs the tool.

While the tool can copy the application and databases, there are some components that it does not support. These components would need to be handled with scripts or manual steps. These include:

- Custom action libraries
- Database tables other than Administrator and Fingerprints
- Any custom files.

A variant approach is to have the source control system place all of the files into the production system and use the application copy tool only to move the administrator and fingerprint databases to production databases.

Parent topic: [Best practices for source control of Datacap applications](#)

Informacje dodatkowe

Informacje referencyjne dotyczące narzędzi, komend instalacyjnych i tworzenia aplikacji.

- [Metody interfejsu REST API w usłudze Datacap Web Services](#)
Datacap Web Services to usługa WWW w środowisku Windows lub usługa WWW oparta na serwerze IIS firmy Microsoft służąca do interakcji z programem Datacap za pomocą prostego, działającego niezależnie od platformy interfejsu programistycznego (API) opartego na protokole REST (Representational State Transfer). Produkt Datacap Web Services obsługuje metody HTTP GET, POST i PUT do tworzenia, aktualizowania i publikowania partii.
- [Narzędzie Fingerprint Maintenance Tool](#)
Wykorzystaj narzędzie Fingerprint Maintenance Tool do przechowywania zsynchronizowanych informacji o odciskach palców we wszystkich miejscach, w których są przechowywane, w tym w bazie danych aplikacji Fingerprint oraz plikach z danymi o odciskach palców.
- [Odwołanie do zmiennej specjalnej parametru Smart](#)
Zmienne specjalne można wykorzystywać razem z parametrami Smart. Parametry Smart nie współpracują ze wszystkimi działaniami.
- [Odwołanie do zmiennej standardowej](#)
Zmienne standardowe można wykorzystywać w partiach, dokumentach, stronach i polach.
- [Odwołanie do zmiennej charakterystycznej dla aplikacji](#)
Niektóre zmienne opracowano wyłącznie z myślą o wykorzystaniu w konkretnych aplikacjach programu Datacap.
- [Podsumowania biblioteki działań](#)
Informacje są dostępne dla wszystkich działań w środowisku Datacap Studio. Aby uzyskać dostęp do wbudowanej pomocy, wybierz działanie na karcie Biblioteki działań i kliknij przycisk „i”.
- [Dodawanie skrótów klawiszowych do aplikacji](#)
Można utworzyć skróty klawiszowe służące do nawigacji i wybierania przycisków interfejsu w obrębie określonej czynności w aplikacji. Aby dodać do aplikacji nowy skrót, należy wyedytować odpowiedni plik konfiguracyjny XML czynności.

FastDoc keyboard shortcuts

FastDoc provides keyboard navigation to make it more accessible to users with limited dexterity, low vision, or other disabilities.

You can use the keyboard to navigate through FastDoc without using a mouse. Use the following table to determine which keys to use. The keys that you use affect the item that currently has focus. Focus is indicated by a dotted line around the item.

To Do This	Do This
Assign the focus to the Active Image pane	CTRL+I
Assign the focus to the Active Stream pane	CTRL+D
Autofit the active image horizontally and vertically	CTRL+1
Autofit the active image horizontally	CTRL+2
Autofit the active image vertically	CTRL+3
Clear the current field	CTRL+Z
Click N Key to capture data from image	Click in the index field in the Verify pane, then click and drag the mouse on the image to draw a new zone. The zone can be empty or it can contain data.

To Do This	Do This
Finish the current task and go to the next available task	CTRL+K CTRL+T
Go to next Low Confidence or error field	ALT+L
Make the current page a trailing page	ALT+CTRL+PgDn
Make the current page the first page of a document	ALT+CTRL+PgUp
Move the image down	SHIFT+ALT+Numeric Keypad 8
Move the image to the left	SHIFT+ALT+Numeric Keypad 6
Move the image to the right	SHIFT+ALT+Numeric Keypad 4
Move the image up	SHIFT+ALT+Numeric Keypad 2
Move the image or pan to the left or right	Click and drag with right mouse button ALT+Right or Left arrows ALT+Click and drag with left mouse button
Move the image or pan up or down	Click and drag with right mouse button ALT+Up or Downarrows ALT+Click and drag with left mouse button
Go to the next image	PgDn
Go to the previous image	PgUp
Place the focus in the Document Type list	ALT+G
Quit the current task and put the batch on hold	CTRL+Q CTRL+S
Run validations	ALT+V
Run validations and display next problem document	CTRL+F
Toggle between Zoom 100% and Autofit both horizontally and vertically	CTRL+0 CTRL+Double-click left mouse button on image Click middle mouse button
Toggle the current image as the first page of the document or as a trailing page	ALT+D
Zoom 100% (view actual size)	CTRL+~

To Do This	Do This
Zoom in to expand the image expands	Double-click image with left mouse button CTRL+I then pull mouse wheel toward you CTRL+Home CTRL+I then CTRL+Up arrow CTRL+ "+/=" key CTRL+Plus
Zoom out to shrink the image shrinks	Double-click image with left mouse button CTRL+I then push mouse wheel away from you CTRL+End CTRL+I then CTRL+Down arrow CTRL+ "_/-" key CTRL+Minus

Datacap Web Services REST API methods

Datacap Web Services is a Windows web service or Microsoft IIS-based web service for interaction with Datacap through a simple, platform-independent, representational state transfer (REST) application programming interface (API). Datacap Web Services supports the HTTP GET, POST, and PUT methods for you to create, update, and release a batch.

Admin

The Admin endpoints are for administrative tasks such as adding users or editing the workflow.

Queue

The Queue endpoints are for working with batches.

Transaction

The Transaction endpoints are for running rules without using Datacap Server and databases.

- [Session/Logon](#)

This method initiates a new session with the Datacap Server and logs on to the Datacap application. The response header includes a `wTmId` cookie and value that must be included in subsequent requests to maintain the session.

- [Session/Logoff](#)

This method logs off the Datacap application and ends the session with the Datacap Server.

- [Session/ChangeUserPassword](#)

You can change your password by using the web services ChangeUserPassword POST method. Your password can be changed by you or an administrator with privileges to change passwords.

- [Rules/Execute](#)

This method runs rules on a batch.

- [Queue/CreateBatch](#)

This method creates a new batch record in the application's Engine Database. The new batch record is uniquely identified by a batch queue ID.

- [Queue/DeleteBatches](#)
This method deletes batches that are associated with an application name.
- [Queue/CheckIntegrity](#)
This method checks the integrity of a batch by using the application name and the queue ID. The page file must be uploaded to the server before you run the CheckIntegrity GET method.
- [Queue/SaveBatchAttribute](#)
This method saves the batch attributes, including extra batch attributes.
- [Queue/GetBatchAttributes](#)
This method returns the attributes of a batch that is associated with a Datacap application name and queue ID.
- [Queue/GetBatchId](#)
This method returns the ID for a batch that is associated with a Datacap application name and queue ID.
- [Queue/GetBatchHistory](#)
This method returns the batch history details of a batch that is associated with the Datacap application name and the queue ID.
- [Queue/GetBatchList](#)
This method returns a list of batches that are based on the application name, page size, page index, sort column, and optional filters.
- [Queue/GrabBatch](#)
This method sets the batch status to `running` for the pending task that is associated with a Datacap application name and queue ID.
- [Queue/GrabNextPendingBatchOnJobTaskList](#)
This method sets the next pending batch status to `running` by using the application, job name list, and task name list. The job name list and the task name list define the job and task pair that is to be used.
- [Queue/ReleaseBatch](#)
This method releases a batch that is associated with a Datacap application name, queue ID, and status. When the current task is complete, the batch is released for processing the next task in the workflow.
- [Queue/GetCCO](#)
This method returns the CCO information for an image file by using the application name, the batch queue ID, and the image file name.
- [Queue/UploadFile](#)
This method uploads a file to a batch folder that is associated with the application name and queue ID. A page object is added to the page file.
- [Queue/SetFile](#)
This method uploads a file or multiple files to a batch folder that is associated with the application name, queue ID, file name, and file extension. A page object is not added to the page file.
- [Queue/GetFile](#)
This method returns a file that is associated with a Datacap application name, queue ID, file name, and file extension.
- [Queue/CopyFilesToCache](#)
This method copies files from the batch folder to the cache folder without downloading the files to the client. When files are uploaded or downloaded, they are placed in the cache. This method is used to run rules on data files such as, `tm000001.xml` and `tm00000.cco`, for pages that were not downloaded.
- [Queue/SetPageFileName](#)
This method assigns a name to the page file for the batch that is associated with a Datacap application, queue ID, file name, and file extension.
- [Queue/GetPageFile](#)
This method returns the contents of the page file for a batch that is associated with a Datacap application name and queue ID.
- [Queue/GetPageFileName](#)
This method returns the name of the page file of a batch that is associated with a Datacap application name and queue ID.

- [Admin/GetApplicationList](#)
This method returns a list of applications that are defined in the Datacap Application Manager.
- [Admin/GetProgramFile](#)
This method returns the requested program settings file that is associated with the application name and the file name. The user must have administrator workflow permission.
- [Admin/SetUserPermissionList](#)
This method saves the job and task index pair list for which the user has permission.
- [Admin/SetGroupPermissionList](#)
This method saves the job and task index pair list for which the group has permission.
- [Admin/GetUserPermissionList](#)
This method returns a job and task index pair list for which the user has permissions.
- [Admin/GetGroupPermissionList](#)
This method returns a job and task index pair list for which the group has permissions.
- [Admin/SaveTask](#)
This method saves task properties such as the application name, description of the task, and program file name. You can queue the batch by a user or station and set the task properties mode for normal processing, batch creation, or task routing.
- [Admin/GetMobileProfiles](#)
This method returns the mobile profiles of the application for which you have permission. Each mobile profile represents one job that is enabled for mobile capture and includes image requirements and batch-level fields that can be entered on the mobile device.
- [Transaction/Start](#)
This method starts a transaction and creates a new transaction ID. With the transaction endpoints, you can run rules without connecting to a Datacap Server, or database.
- [Transaction/SetFile](#)
This method uploads a file to a transaction by using the transaction ID, file name, and file extension. A page object is not added to the page file.
- [Transaction/Execute](#)
This method runs rules for a transaction.
- [Transaction/GetFile](#)
This method returns a file in a transaction by using the transaction ID, file name, and file extension.
- [Transaction/GetFileList](#)
This method gets a list of files in a transaction by using the transaction ID and mode.
- [Transaction/End](#)
This method ends the transaction and removes files that are associated with the transaction ID.

Related concepts:

[Datacap Web Services authentication](#)
[Datacap Web Services installation steps](#)

Session/Logon

This method initiates a new session with the Datacap Server and logs on to the Datacap application. The response header includes a `wTmId` cookie and value that must be included in subsequent requests to maintain the session.

The session timeout is specified in `wtm\web.config` file. If the number of minutes is exceeded, the session expires and the user must log on again.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Session/Logon`

Request content

The request for this method contains no content.

Response content

The `Set-Cookie` session value `wTmId=<session cookie>` that is returned by this method is kept within the session header of subsequent requests.

This method returns one of the following response codes.

Table 1. Response codes for the POST Logon method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

POST Logon method JSON request example

```
{
  "application": "String content",
  "password": "String content",
  "station": "String content",
  "user": "String content"
}
```

POST Logon method XML request example

```
<LogonProperties>
  <application>String content</application>
  <password>String content</password>
  <station>String content</station>
  <user>String content</user>
</LogonProperties>
```

Parent topic: [Datacap Web Services REST API methods](#)

Session/Logoff

This method logs off the Datacap application and ends the session with the Datacap Server.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Session/Logoff`

Request content

The request content includes the `wTmId` cookie the request header.

Response content

The response for this method contains no content.

This method returns one of the following response codes.

Table 1. Response codes for the POST Logoff method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

POST Logoff method JSON request example

```
{
  "application": "String content",
  "password": "String content",
  "station": "String content",
  "user": "String content"
}
```

POST Logoff method XML request example

```
<LogoffProperties>
  <application>String content</application>
  <password>String content</password>
  <station>String content</station>
  <user>String content</user>
</LogoffProperties>
```

Parent topic: [Datacap Web Services REST API methods](#)

Session/ChangeUserPassword

You can change your password by using the web services `ChangeUserPassword` POST method. Your password can be changed by you or an administrator with privileges to change passwords.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Session/ChangeUserPassword`

Request content

The request contains application, password, and user information.

Response content

The response contains the change result with a true/false string in the *changeResult* attribute. The response also provides a message that states the change is complete or the change failed.

This method returns one of the following response codes.

Table 1. Response codes for the POST ChangeUserPassword method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

POST ChangeUserPassword method JSON request example

```
{
  "application": "String content",
  "newPassword": "String content",
  "oldPassword": "String content",
  "user": "String content"
}
```

POST ChangeUserPassword method XML request example

```
<ChangePasswordProperties>
  <application>String content</application>
  <newPassword>String content</newPassword>
  <oldPassword>String content</oldPassword>
  <user>String content</user>
</ChangePasswordProperties>
```

POST ChangeUserPassword method JSON response example

```
{
  "changeResult": true/false,
  "msg": "String content"
}
```

POST ChangeUserPassword method XML response example

```
<ChangePasswordResult>
  <changeResult>true</changeResult>
  <msg>String content</msg>
</ChangePasswordResult>
```

Parent topic: [Datacap Web Services REST API methods](#)

Rules/Execute

This method runs rules on a batch.

Files that are retrieved from the batch folder or sent to the batch folder are cached in a session folder on the Datacap Web Services server. When the web services session ends, the session folder is deleted. When the application closes, all folders are deleted.

The Execute POST method does not update batch attributes. If one of the values is changed by the rules, such as *Priority*, the batch attributes must be updated by using *SaveBatchAttribute*.

Caching is not required to run rules. If the *BatchDir* value is specified, the Datacap Web Services accesses the batch folder directly. The *BatchDir* value must be empty when the web.config setting *enableCache* is set to True to run rules against the files in the cache folder. If *enableCache* is set to False, you must specify a batch folder in *BatchDir*.

- *Result* is not needed when you are posting. It is the result of rules execution. You can post with `<Result></Result>`.
- If you post the *TargetDCOObject* property as empty, the page is targeted.
- *ChildCondition* is not needed when you are posting. It is the result of rules execution that is set by the action *Task_RaiseCondition*. *ChildCondition* can be empty or omitted when you post.
- *ChildrenQuantity* is not needed when you are posting. It is the result of rules execution as set by the action *Task_NumberOfSplits*. *ChildrenQuantity* can be empty or omitted when you post.
- *XtraBatchFields* are custom fields added to the Job Monitor. See *Creating a custom column in the Job Monitor*. The caller can use the *SaveBatchAttribute* endpoint to update the custom fields in the database after the actions change them. *XtraBatchFields* can be empty or omitted when you post. If actions need access to extra batch fields, the caller must pass *XtraBatchFields*, as in the following example.

```
<XtraBatchFields>
  <KeyValueOfstringstring
    xmlns=""http://schemas.microsoft.com/2003/10/Serialization/Arrays"">
    <Key>pb_FieldA</Key>
    <Value>ValueA</Value>
  </KeyValueOfstringstring>
</XtraBatchFields>
```

- *AllowOverride* means that the actions are allowed to override failed validations. *AllowOverride* can be omitted when you are posting. If *AllowOverride* is included, the value must be set but it is not used. In the response, *AllowOverride* is set to False, if the action *SetIsOverrideable* is called with a parameter of *False* and a validation failed.
- *DocsInBatch* is the number of documents in the batch, which you can find in the page file. *DocsInBatch* can be used by actions.
- *PagesInBatch* is the number of pages in the batch, which you can find in the page file. *PagesInBatch* can be used by actions.
- *AppName* (application name) and *Workflow* (workflow name) are not always the same.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Rules/Execute`

Request content

The request content for this method runs the rules on a batch.

Response content

The response for this method contains the same content as the Request content.

This method returns one of the following response codes.

Table 1. Response codes for the POST Execute method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

POST Execute method JSON request example

```
{
  "Profile": "Verify",
  "Properties": {
    "AllowOverride": true,
    "AppName": "Flex",
    "BatchDir": "C:\\DatacapFlex\\batches20140804.000000",
    "BatchID": "20140804.000000",
    "DCOFile": "Verify.xml",
    "DocsInBatch": 1,
    "JobName": "VSan-TIFF",
    "Operator": "admin",
    "PagesInBatch": 1,
    "Priority": 1,
    "Station": "1",
    "TaskName": "Verify",
    "Workflow": "Flex"
  }
}
```

POST Execute method XML request example

```
<RulesProperties>
  <Profile>Verify</Profile>
  <Properties>
    <AllowOverride>true</AllowOverride>
    <AppName>Flex</AppName>
    <BatchDir>C:\\Datacap\\Flex\\batches\\20140804.000000</BatchDir>
    <BatchID>20140804.000000</BatchID>
    <DCOFile>verify.xml</DCOFile>
    <DocsInBatch>1</DocsInBatch>
    <JobName>VScan-TIFF</JobName>
    <Operator>admin</Operator>
    <PagesInBatch>1</PagesInBatch>
    <Priority>1</Priority>
    <Station>1</Station>
    <TaskName>Verify</TaskName>
    <Workflow>Flex</Workflow>
  </Properties>
</RulesProperties>
```

```
</Properties>
</RulesProperties>
```

Parent topic: [Datacap Web Services REST API methods](#)

Related reference:

[Queue/SaveBatchAttribute](#)

Related information:

[Creating a custom column in the Job Monitor](#)

[Task_NumberOfSplits](#)

[Task_RaiseCondition](#)

[SetIsOverrideable](#)

Queue/CreateBatch

This method creates a new batch record in the application's Engine Database. The new batch record is uniquely identified by a batch queue ID.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Queue/CreateBatch`

Request content

The request contains the application name and the job name. The request can also contain the batch folder name, the new page file name, and the name of an existing page file.

Table 1. Request content attributes for the CreateBatch POST method

Attribute Name	Description
<i>application</i>	The Datacap application name that is shown in the datacap.xml file. The application name is required and must be the first attribute.
<i>job</i>	The job name that is shown in a Datacap application workflow. The job name is required and the batch creation task must be the first task in the job.
<i>batchFolder</i>	The name of the batch folder that is created is optional. When this variable is not included, the batch folder is assigned a name by Datacap.
<i>pageFile</i>	The name of the page file that is created is optional. When this variable is not included, the page file is assigned a name by Datacap. Do not include a <i>pageFile</i> attribute with a blank value. The following are the <i>pageFile</i> value options. <ul style="list-style-type: none">• The string to use as the file name.• A file attribute that specifies an existing page file.• The XML content of the page file.
<i>file</i>	The name of an existing page file is optional. When you include this attribute, you must embed it as a child node of <i>pageFile</i> and provide the full path, file name, and file extension. When you use this variable, you must also include the <i>batchFolder</i> variable.

Response content

The response for this method returns the following batch attributes.

Table 2. Response content attributes for the CreateBatch POST method

Attribute name	Description of value
<i>batchID</i>	The batch ID that is assigned to the batch by Datacap.
<i>job</i>	The name of the job.
<i>queueId</i>	The unique identifier that is assigned to the batch by Datacap.
<i>status</i>	The status of the batch.
<i>task</i>	The name of the task.

This method returns one of the following response codes.

Table 3. Response codes for the CreateBatch POST method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

CreateBatch POST method XML request example 1

Datacap creates the batch folder and manages the adding of each file to the DCO page file, such as adding the *tm000001* node to scan.xml. The request content contains an empty text value for the *batchFolder* node. The message content does not contain a *pageFile* node.

```
<createBatchAttributes>
  <application>APT</application>
  <batchFolder></batchFolder>
  <job>Main Job</job>
</createBatchAttributes>
```

CreateBatch POST method XML request example 2

Datacap creates the batch folder and the client manages the DCO page file. The request content *batchFolder* node contains the XML to use for the page file. The *B id* batch node attribute must be blank. This method is useful for transferring metadata with or without the page objects. If page objects are specified, the UploadFile endpoint must be used instead of SetFile.

```
<createBatchAttributes>
  <application>APT</application>
```

```

    <batchFolder></batchFolder>
    <job>Main Job</job>
    <pageFile>
<B
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.install.doc_dcdws034_">
  <V n="STATUS">0</V>
  <V n="TYPE">FastDoc</V>
  <P
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.install.doc_dcdws034_TM00
0001">
    <V n="TYPE">Other</V>
    <V n="Status">49</V>
    <V n="IMAGEFILE">TM000001.tif</V>
  </P>
</B>
</pageFile>
</createBatchAttributes>

```

CreateBatch POST method XML request example 3

The client creates the batch folder and manages the DCO page file. The page file name is provided and the page file is separately uploaded. The request content message body *pageFile* node text contains the name of the page file, such as scan.xml. After CreateBatch is called, then SetFile is called to upload the page file.

```

<createBatchAttributes>
  <application>APT</application>
  <batchFolder>C:\Datacap\Batches\20130328.000087</batchFolder>
  <job>Main Job</job>
  <pageFile><file>scan.xml</file></pageFile>
</createBatchAttributes>

```

Parent topic: [Datacap Web Services REST API methods](#)

Queue/DeleteBatches

This method deletes batches that are associated with an application name.

URI

http://{IP address}:{Port}/ServicewTM.svc/Queue/DeleteBatches/{*application*}

The URI for this method includes the following path elements.

Table 1. Path elements for the POST DeleteBatches method

Name	Type	Required?	Description
{ <i>application</i> }	String	Yes	The name of the application for which batches are to be deleted.

Request content

The request deletes the batches that are associated with the application name.

Response content

The response for this method contains no content.

This method returns one of the following response codes.

Table 2. Response codes for the POST DeleteBatches method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/CheckIntegrity

This method checks the integrity of a batch by using the application name and the queue ID. The page file must be uploaded to the server before you run the CheckIntegrity GET method.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Queue/CheckIntegrity/{application}/{queueId}`

The URI for this method includes the following path elements.

Table 1. Path elements for the GET CheckIntegrity method

Name	Type	Required?	Description
<code>{application}</code>	String	Yes	The name of the application for which the batch integrity is checked.
<code>{queueId}</code>	String	Yes	The unique identifier that is assigned to the batch.

Request content

The request for this method contains no content.

Response content

The response for this method returns the ID, message, and returnCode content.

ID - In case of error, the last object that is checked.

message - Return message.

returnCode - One of the following values is returned.

0 = Passed.

1 = Has more child objects that allowed by the maximum attribute.

2 = Has fewer child objects than required by the minimum attribute.

3 = Invalid member – a child object is not of a type that is supported by the parent.

4 = A child object is in the wrong position relative to other child objects as specified by the position attribute.

This method also returns one of the following response codes.

Table 2. Response codes to the GET CheckIntegrity method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service due to malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

GET CheckIntegrity method JSON response example

```
{
  "ID": "String content",
  "message": "String content",
  "returnCode": 2147483647
}
```

GET CheckIntegrity method XML response example

```
<CheckIntegrityResult>
  <ID>String content</ID>
  <message>String content</message>
  <returnCode>2147483647</returnCode>
</CheckIntegrityResult>
```

Parent topic: [Datacap Web Services REST API methods](#)

Queue/SaveBatchAttribute

This method saves the batch attributes, including extra batch attributes.

- The logon user must have Status change/rollback privileges for the jobTaskOrder, queueStatus, and taskID attributes to be saved.
- The logon user must have Priority/Operator change privileges for the QueuePriority, Operator, Station, SkipOperator, or SkipStation attributes to be saved.
- The logon user must have Batch attributes change privileges for the BatchDir or PageFile attributes to be saved.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Queue/SaveBatchAttribute/{application}`

The URI for this method includes the following path elements.

Table 1. Path elements for the POST SaveBatchAttribute method

Name	Type	Required?	Description
<i>{application}</i>	String	Yes	The name of the application for which the batch attributes are to be saved.

Request content

The request content saves the batch attributes.

Response content

The response return the message and returnCode content.

This method returns one of the following response codes.

Table 2. Response codes for the POST SaveBatchAttribute method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

POST SaveBatchAttribute method JSON request example

```
{
  "batchDir": "String content",
  "jobTaskOrder": 2147483647,
  "operationUser": "String content",
  "pageFile": "String content",
  "queueID": 2147483647,
  "queuePriority": 2147483647,
  "queueStatus": "String content",
  "skipOperator": "String content",
  "skipStation": "String content",
  "station": "String content",
  "taskID": "String content",
  "xtraBatchFields": {
    "Count": 2147483647,
    "Fields": [
      {
        "field": "String content",
        "value": "String content"
      }
    ]
  }
}
```

```
}  
}
```

POST SaveBatchAttribute method XML request example

```
<BatchAttrSave>  
  <batchDir>String content</batchDir>  
  <jobTaskOrder>2147483647</jobTaskOrder>  
  <operationUser>String content</operationUser>  
  <pageFile>String content</pageFile>  
  <queueID>2147483647</queueID>  
  <queuePriority>2147483647</queuePriority>  
  <queueStatus>String content</queueStatus>  
  <skipOperator>String content</skipOperator>  
  <skipStation>String content</skipStation>  
  <station>String content</station>  
  <taskID>String content</taskID>  
  <xtraBatchFields>  
    <Count>2147483647</Count>  
    <Fields>  
      <XtraBatchField>  
        <field>String content</field>  
        <value>String content</value>  
      </XtraBatchField>  
      <XtraBatchField>  
        <field>String content</field>  
        <value>String content</value>  
      </XtraBatchField>  
    </Fields>  
  </xtraBatchFields>  
</BatchAttrSave>
```

POST SaveBatchAttribute method JSON response example

```
{  
  "message": "String content",  
  "returnCode": 2147483647  
}
```

POST SaveBatchAttribute method XML response example

```
<ReturnResult>  
  <message>String content</message>  
  <returnCode>2147483647</returnCode>  
</ReturnResult>
```

Parent topic: [Datacap Web Services REST API methods](#)

Queue/GetBatchAttributes

This method returns the attributes of a batch that is associated with a Datacap application name and queue ID.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Queue/GetBatchAttributes/{application}/{queueId}`

The URI for this method includes the following path elements.

Table 1. Path elements for the GET GetBatchAttributes method

Name	Type	Required?	Description
<i>{application}</i>	String	Yes	The name of the application for which the batch attributes are to be returned.
<i>{queueId}</i>	String	Yes	The unique identifier that is assigned to the batch.

Request body

The request for this method contains no content.

Response content

The response contains the following batch attributes.

Table 2. Response content for the GET GetBatchAttributes method

Attribute Name	Description
<i>batchID</i>	The batch ID that is assigned to the batch.
<i>job</i>	The name of the job.
<i>queueID</i>	The unique identifier that is assigned to the batch.
<i>status</i>	The status of the batch.
<i>task</i>	The name of the task.

This method also returns one of the following response codes.

Table 3. Response codes for the GET GetBatchAttributes method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/GetBatchId

This method returns the ID for a batch that is associated with a Datacap application name and queue ID.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Queue/GetBatchId/{application}/{queueId}`

The URI for the GET method includes the following path elements.

Table 1. Path elements for the GetBatchId GET method

Name	Type	Required?	Description
<i>{application}</i>	String	Yes	The name of the application for which the batch ID is to be returned.
<i>{queueId}</i>	String	Yes	The unique identifier that is assigned to the batch.

Request body

The request for this method contains no content.

Response content

The response contains the batch ID.

This method also returns one of the following response codes.

Table 2. Response codes for the GET GetBatchId method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/GetBatchHistory

This method returns the batch history details of a batch that is associated with the Datacap application name and the queue ID.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Queue/GetBatchHistory/{application}/{queueId}`

The URI for this method includes the following path elements.

Table 1. Path elements for the GET GetBatchHistory method

Name	Type	Required?	Description
<i>{application}</i>	String	Yes	The name of the application for which the batch history is to be returned.
<i>{queueId}</i>	String	Yes	The unique identifier that is assigned to the batch.

Request body

The request for this method contains no content.

Response content

The response contains the batch history details.

This method also returns one of the following response codes.

Table 2. Response codes for the GET GetBatchHistory method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/GetBatchList

This method returns a list of batches that are based on the application name, page size, page index, sort column, and optional filters.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Queue/GetBatchList/{application}/{pageSize}/{pageIndex}/{sort}`

The URI for this method includes the following path elements.

Table 1. Path elements for the GET GetBatchList method

Name	Type	Required?	Description
<code>{application}</code>	String	Yes	The name of the application for which the list of batches is to be returned.
<code>{pageSize}</code>	String	Yes	The maximum number of batches in a list that you can retrieve. If you set the <code>pageSize</code> value to 0 to return a list of all batches, you must include optional filtering to avoid a performance impact on the Datacap Server and Datacap Web Services.
<code>{pageIndex}</code>	String	Yes	The specific batches in a list that you want to retrieve. For example, if there are 30 batches and your <code>pageSize</code> is 10, enter 0 to retrieve the list of the first 10 batches. Enter 1 to retrieve the list of the second 10 batches and enter 2 to retrieve the list of the last 10 batches.
<code>{sort}</code>	String	Yes	You can sort the list of batches by any column in the response.

Name	Type	Required?	Description
Optional filters	String	No	When you add an optional filter, you must use the format, databaseField=Operator Value. If you add more than one optional filter, you must put an ampersand between the filters. In the following example, this method returns a list of the first 10 batches that have a pending status and a priority value of 1. "http://localhost:55287/ServicewTM.svc/Queue/GetBatchList/a/ppname/10/0/qu_id?qu_status== pending&qu_priority== 1 Date filtering is not supported.

Request content

The request for this method contains no content.

Response content

The response for this method returns a list of batches.

This method also returns one of the following response codes:

Table 2. Response codes to the GET GetBatchList method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service due to malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the Request-Line is not allowed for the resource that is identified by the Request-URI. The response must include an Allow header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

GET GetBatchList method JSON response example

```
{
  "Count":2147483647,
  "Batches":[{
    "pb_StationID":"String content",
    "pb_UserID":"String content",
    "pb_adjustdocs":"String content",
    "pb_adjustpages":"String content",
    "pb_batch":"String content",
    "pb_batchdir":"String content",
    "pb_expectdocs":"String content",
    "pb_expectpgs":"String content",
    "pb_headertable":"String content",
    "pb_mrdate":"String content",
    "pb_ndocs":"String content",
    "pb_needMeet":"String content",
    "pb_pagefile":"String content",
```

```

    "pb_pages":"String content",
    "pb_parentbatch":"String content",
    "qs_OpSkip":"String content",
    "qs_StSkip":"String content",
    "qs_elaps":"String content",
    "qs_op":"String content",
    "qs_qid":"String content",
    "qs_start":"String content",
    "qs_station":"String content",
    "qs_stop":"String content",
    "qs_taskid":"String content",
    "qs_tsorder":"String content",
    "qu_admDB":"String content",
    "qu_batch":"String content",
    "qu_counter":"String content",
    "qu_done":"String content",
    "qu_elaps":"String content",
    "qu_id":"String content",
    "qu_job":"String content",
    "qu_lock":"String content",
    "qu_parent":"String content",
    "qu_priority":"String content",
    "qu_source":"String content",
    "qu_spawnstype":"String content",
    "qu_start":"String content",
    "qu_status":"String content",
    "qu_task":"String content",
    "qu_tsorder":"String content",
    "xtraBatchFields":{
      "Count":2147483647,
      "Fields":[{
        "field":"String content",
        "value":"String content" }] }
  ]]
}

```

GET GetBatchList method XML response example

```

<BatchList> <Count>2147483647</Count>
  <Batches>
    <Batch>
      <pb_StationID>String content</pb_StationID>
      <pb_UserID>String content</pb_UserID>
      <pb_adjustdocs>String content</pb_adjustdocs>
      <pb_adjustpages>String content</pb_adjustpages>
      <pb_batch>String content</pb_batch>
      <pb_batchdir>String content</pb_batchdir>
      <pb_expectdocs>String content</pb_expectdocs>
      <pb_expectpgs>String content</pb_expectpgs>
      <pb_headertable>String content</pb_headertable>
      <pb_mrdate>String content</pb_mrdate>
      <pb_ndocs>String content</pb_ndocs>
      <pb_needMeet>String content</pb_needMeet>
      <pb_pagefile>String content</pb_pagefile>
      <pb_pages>String content</pb_pages>
      <pb_parentbatch>String content</pb_parentbatch>
      <qs_OpSkip>String content</qs_OpSkip>
      <qs_StSkip>String content</qs_StSkip>
      <qs_elaps>String content</qs_elaps>
      <qs_op>String content</qs_op>
      <qs_qid>String content</qs_qid>
      <qs_start>String content</qs_start>
      <qs_station>String content</qs_station>
    </Batch>
  </Batches>
</BatchList>

```

```

<qs_stop>String content</qs_stop>
<qs_taskid>String content</qs_taskid>
<qs_tsorder>String content</qs_tsorder>
<qu_admDB>String content</qu_admDB>
<qu_batch>String content</qu_batch>
<qu_counter>String content</qu_counter>
<qu_done>String content</qu_done>
<qu_elaps>String content</qu_elaps>
<qu_id>String content</qu_id>
<qu_job>String content</qu_job>
<qu_lock>String content</qu_lock>
<qu_parent>String content</qu_parent>
<qu_priority>String content</qu_priority>
<qu_source>String content</qu_source>
<qu_spawntype>String content</qu_spawntype>
<qu_start>String content</qu_start>
<qu_status>String content</qu_status>
<qu_task>String content</qu_task>
<qu_tsorder>String content</qu_tsorder>
<xtraBatchFields>
  <Count>2147483647</Count>
  <Fields>
    <XtraBatchField>
      <field>String content</field>
      <value>String content</value>
    </XtraBatchField>
    <XtraBatchField>
      <field>String content</field>
      <value>String content</value>
    </XtraBatchField>
  </Fields>
</xtraBatchFields>
</Batch>
</Batches>
</BatchList>

```

Parent topic: [Datacap Web Services REST API methods](#)

Queue/GrabBatch

This method sets the batch status to `running` for the pending task that is associated with a Datacap application name and queue ID.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Queue/GrabBatch/{application}/{queueId}`

The URI for this method includes the following path elements.

Table 1. Path elements for the PUT GrabBatch method

Name	Type	Required?	Description
<code>{application}</code>	String	Yes	The name of the application for which the batch status is set to <code>running</code> .
<code>{queueId}</code>	String	Yes	The unique identifier that is assigned to the batch.

Request content

The request for this method contains no content.

Response content

The response sets the batch status to `running` for the pending task.

This method returns one of the following response codes.

Table 2. Response codes for the PUT GrabBatch method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/GrabNextPendingBatchOnJobTaskList

This method sets the next pending batch status to `running` by using the application, job name list, and task name list. The job name list and the task name list define the job and task pair that is to be used.

URI

`http://{IP address}:`
`{Port}/ServicewTM.svc/Queue/GrabNextPendingBatchOnJobTaskList/{application}/{jobNameList}/{taskNameList}`

The URI for this method includes the following path elements.

Table 1. Path elements for the PUT method

Name	Type	Required?	Description
<code>{application}</code>	String	Yes	The name of the application for which the next pending batch status is to be set.
<code>{jobNameList}</code>	String	Yes	The list of job names that are separated by a comma, such as <code>Main Job, Web Job</code> .
<code>{taskNameList}</code>	String	Yes	The list of task names that are separated by a comma, such as <code>VScan, Verify, Export</code>

Request content

The request for this method contains no content.

Response content

The response sets the status of the next pending batch on the job and task list to `running`.

This method also returns one of the following response codes.

Table 2. Response codes for the PUT GrabNextPendingBatchOnJobTaskList method

Response Code	Description
201 Created	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/ReleaseBatch

This method releases a batch that is associated with a Datacap application name, queue ID, and status. When the current task is complete, the batch is released for processing the next task in the workflow.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Queue/ReleaseBatch/{application}/{queueId}/{status}`

The URI for this method includes the following path elements.

Table 1. Path elements for the PUT ReleaseBatch method

Name	Type	Required?	Description
<code>{application}</code>	String	Yes	The name of the application for which the batch is released.
<code>{queueId}</code>	String	Yes	The unique identifier that is assigned to the batch.
<code>{status}</code>	String	Yes	The following are the value options for <code>status</code> . <ul style="list-style-type: none">• hold• finished• canceled• aborted• offline

Request content

Table 2. Request content attributes for the PUT ReleaseBatch method

Attribute Name	Description
<code>pageFile</code>	Optional. The XML contents of the page file for the batch.

For examples of how to specify the XML contents for the `pageFile` attribute, see [Queue/CreateBatch](#).

Response content

The response releases the batch for the next task in the workflow to process.

This method returns one of the following response codes.

Table 3. Response codes for the PUT ReleaseBatch method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/GetCCO

This method returns the CCO information for an image file by using the application name, the batch queue ID, and the image file name.

URI

`http://{IP address}:{Port}/ServiceWtm.svc/Queue/GetCCO/{application}/{queueId}/{imageFilename}`

The URI for this method includes the following path elements.

Table 1. Path elements for the GET GetCCO method

Name	Type	Required?	Description
<code>{application}</code>	String	Yes	The name of the application for which the CCO information is returned.
<code>{queueId}</code>	String	Yes	The unique identifier that is assigned to the batch.
<code>{imageFilename}</code>	String	Yes	The name of the image file. You must not include the file name extension.

Request content

The request for this method contains no content.

Response content

The response content is the CCO information for the image file.

This method also returns one of the following response codes.

Table 2. Response codes for the GET GetCCO method

Response Code	Description
201 Created	The request was fulfilled and resulted in the creation of a new resource.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included authorization credentials, then the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

GET GetCCO method JSON response example

```
{
  "Count": 2147483647,
  "Lines": [
    {
      "Count": 2147483647,
      "l": 2147483647,
      "t": 2147483647,
      "b": 2147483647,
      "r": 2147483647,
      "Words": [
        {
          "b": 2147483647,
          "l": 2147483647,
          "n": 2147483647,
          "r": 2147483647,
          "t": 2147483647,
          "text": "String content"
        }
      ]
    }
  ]
}
```

GET GetCCOmethod XML response example

```
<Count>2</Count>
<Lines>
  <CCOLine>
    <Count>1</Count><l>1483</l><t>301</t><b>351</b><r>1755</r>
    <Words>
      <CCOWord>
        <b>351</b><l>1483</l><n>0</n><r>1755</r><t>301</t>
        <text>INVOICE</text>
      </CCOWord>
    </Words>
  </CCOLine>
  <CCOLine>
    <Count>3</Count><l>2018</l><t>457</t><b>483</b><r>2272</r>
    <Words>
      <CCOWord>
        <b>483</b><l>2018</l><n>1</n><r>2145</r><t>457</t>
        <text>INVOICE</text>
      </CCOWord>
    </Words>
  </CCOLine>
</Lines>
```

```

    <CCOWord>
      <b>482</b><l>2152</l><n>2</n><r>2172</r><t>458</t>
      <text>#</text>
    </CCOWord>
    <CCOWord>
      <b>483</b><l>2178</l><n>3</n><r>2272</r><t>458</t>
      <text>28100</text>
    </CCOWord>
  </Words>
</CCOLine>
</Lines>
</CCO>

```

Parent topic: [Datacap Web Services REST API methods](#)

Queue/UploadFile

This method uploads a file to a batch folder that is associated with the application name and queue ID. A page object is added to the page file.

Files can be transferred as a binary string or by using Base64 encoding. When Base64 encoding is used, you must add `Content-Transfer-Encoding: base64` to the header when you are streaming, or to the form part for multipart/form-data. For Base64 encoding, you can specify the `Content-Type`: as the file type. The UploadFile request can accept a byte stream as the message body or the message body can use the multipart/form-data format where the file byte stream is contained in each part.

The sequentially numbered page object is created in the page file. The task settings contain the page file name. The page *TYPE* is always set to *Other*, and *STATUS* is always set to 49. The page object contains the following information.

```

<P
id="__dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.install.doc_dcdws044_tm00
0001">
  <V n="TYPE">Other</V>
  <V n="STATUS">49</V>
  <V n="IMAGEFILE">tm000001.tif</V>
</P>

```

You can adjust the value of the `maxAllowedContentLength` and `maxRequestLength` settings in the `web.config` file. The maximum size message that can be uploaded depends on your web server, proxy server, and client.

URI

`http://{IP address}:{Port}/ServiceWtm.svc/Queue/UploadFile/{application}/{queueId}`

The URI for this method includes the following path elements.

Table 1. Path elements for the POST UploadFile method

Name	Type	Required?	Description
<code>{application}</code>	String	Yes	The name of the application for which the file is to be uploaded.
<code>{queueId}</code>	String	Yes	The unique identifier that is assigned to the batch.

Request content

The request content contains a multipart/form-data message, including the file or files that are to be uploaded.

Response content

The response returns the original file name and the page ID.

This method returns one of the following response codes.

Table 2. Response codes for the POST UploadFile method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/SetFile

This method uploads a file or multiple files to a batch folder that is associated with the application name, queue ID, file name, and file extension. A page object is not added to the page file.

Files can be transferred as a binary string or by using Base64 encoding. When Base64 encoding is used, you must add `Content-Transfer-Encoding: base64` to the header when you are streaming, or to the form part for multipart/form-data. For Base64 encoding, you can specify the `Content-Type`: as the file type. The SetFile request can accept a byte stream as the message body or can have the message body use the multipart/form-data format where the file byte stream is contained in each part.

If you specify a header field file name, then it is used as the file name for the uploaded the file. If you do not specify a header field file name, then the `{fileName}` in the SetFile URI is used as the file name for the uploaded file.

You can adjust the value of the `maxAllowedContentLength` and `maxRequestLength` settings in the `web.config` file. The maximum size message that can be uploaded depends on your web server, proxy server, and client.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Queue/SetFile/{application}/{queueId}/{fileName}/{fileExtension}`

The URI for this method includes the following path elements.

Table 1. Path elements for the POST SetFile method

Name	Type	Required?	Description
------	------	-----------	-------------

Name	Type	Required?	Description
{application}	String	Yes	The name of the application for which the file is to be uploaded.
{fileName}	String	Yes	The name of the file that is to be uploaded.
{fileExtension}	String	Yes	The file name extension of the file that is to be returned, such as exe. Do not include the file name path, set.xml.

Request content

The request content contains file that is to be uploaded.

Response content

The response for this method contains no content.

This method returns one of the following response codes.

Table 2. Response codes for the POST SetFile method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/GetFile

This method returns a file that is associated with a Datacap application name, queue ID, file name, and file extension.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Queue/GetFile/{application}/{queueId}/{fileName}/{fileExtension}`

The URI for this method includes the following path elements.

Table 1. Path elements for the GET GetFile method

Name	Type	Required?	Description
{application}	String	Yes	The name of the application for which the file is to be returned.

Name	Type	Required?	Description
<i>{queueId}</i>	String	Yes	The unique identifier that is assigned to the batch.
<i>{fileName}</i>	String	Yes	The name of the file in the batches directory that is to be returned. The <i>{fileName}</i> must not have a dot such as verify.1 or a plus sign such as batch+profiler.
<i>{fileExtension}</i>	String	Yes	The file name extension of the file that is to be returned, such as xml. Do not include the file name path, such as verify.xml.

Request body

The request for this method contains no content.

Response content

The response returns the file.

This method returns one of the following response codes.

Table 2. Response codes for the GET GetFile method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/CopyFilesToCache

This method copies files from the batch folder to the cache folder without downloading the files to the client. When files are uploaded or downloaded, they are placed in the cache. This method is used to run rules on data files such as, tm000001.xml and tm00000.cco, for pages that were not downloaded.

If the file is cached and an endpoint is used to retrieve a file from the batch folder, the cached file is returned. If the file is not in the cache, the file is returned from the batch folder and added to the cache. When you are calling an endpoint that adds or updates a batch file, the file is saved to the cache but it is not saved to the batch folder. When the batch is released, all files are copied from the cache and will write over files that are in the batch folder.

The `wTmId` header value that is returned by the POST method `Session/Logon` must be in the session header.

URI

http://{IP address}:{Port}/ServiceWtm.svc/Queue/CopyFilesToCache/{application}/{queueId}

The URI for this method includes the following path elements.

Table 1. Path elements for the POST CopyFilesToCache method

Name	Type	Required?	Description
{application}	String	Yes	The name of the application for which the files are to be copied to the cache folder.
{queueId}	String	Yes	The unique identifier that is assigned to the batch.

Request content

The request content is the message body.

Response content

The response for this method contains no content.

This method returns one of the following response codes.

Table 2. Response codes for the POST CopyFilesToCache method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

POST CopyFilesToCache method JSON request example

```
["String content"]
```

POST CopyFilesToCache method XML request example

```
<ArrayOfstring xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">  
  <string>tm000001.tif</string>  
  <string>tm000001.cco</string>  
</ArrayOfstring>
```

Parent topic: [Datacap Web Services REST API methods](#)

Queue/SetPageFileName

This method assigns a name to the page file for the batch that is associated with a Datacap application, queue ID, file name, and file extension.

URI

http://{IP address}:
{Port}/ServicewTM.svc/Queue/SetPageFileName/{application}/{queueId}/{fileName}/{fileExtension}

The URI for this method includes the following path elements.

Table 1. Path elements for the PUT SetPageFileName method

Name	Type	Required?	Description
{application}	String	Yes	The name of the application for which the page file name for the batch is assigned.
{queueId}	String	Yes	The unique identifier that is assigned to the batch.
{fileName}	String	Yes	The name of the file that is assigned.
{fileExtension}	String	Yes	The file name extension of the file that is to be returned, such as exe. Do not include the file name path, set.xml.

Request content

The request for this method contains no content.

Response content

The response sets the page file name for the batch.

This method returns one of the following response codes.

Table 2. Response codes for the PUT SetPageFileName method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/GetPageFile

This method returns the contents of the page file for a batch that is associated with a Datacap application name and queue ID.

URI

http://{IP address}:{Port}/ServiceWtm.svc/Queue/GetPageFile/{application}/{queueId}

The URI for this method includes the following path elements.

Table 1. Path elements for the GET GetPageFile method

Name	Type	Required?	Description
{application}	String	Yes	The name of the application for which the contents of the page file are to be returned.
{queueId}	String	Yes	The unique identifier that is assigned to the batch.

Request body

The request for this method contains no content.

Response content

The response contains the XML contents of the page file.

This method returns one of the following response codes.

Table 2. Response codes for the GET GetPageFile method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the Request-Line is not allowed for the resource that is identified by the Request-URI. The response must include an Allow header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Queue/GetPageFileName

This method returns the name of the page file of a batch that is associated with a Datacap application name and queue ID.

URI

http://{IP address}:{Port}/ServiceWtm.svc/Queue/GetPageFileName/{application}/{queueId}

The URI for this method includes the following path elements.

Table 1. Path elements for the GET GetPageFileName method

Name	Type	Required?	Description
{application}	String	Yes	The name of the application for which the page file name is to be returned.
{queueId}	String	Yes	The unique identifier that is assigned to the batch.

Request body

The request for this method contains no content.

Response content

The response contains the page file name.

This method returns one of the following response codes.

Table 2. Response codes for the GET GetPageFileName method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Admin/GetApplicationList

This method returns a list of applications that are defined in the Datacap Application Manager.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Admin/GetApplicationList`

Request content

The request for this method contains no content.

Response content

The response contains the list of applications that are defined in the Datacap Application Manager.

This method also returns one of the following response codes.

Table 1. Response codes for the GET GetApplicationList method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.

Response Code	Description
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

GET GetApplicationList method JSON response example

```
{ "Applications":["String content"] }
```

GET GetApplicationList method XML response example

```
<ApplicationList>
  <Applications>
    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">String
content</string>
    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">String
content</string>
  </Applications>
</ApplicationList>
```

Parent topic: [Datacap Web Services REST API methods](#)

Admin/GetProgramFile

This method returns the requested program settings file that is associated with the application name and the file name. The user must have administrator workflow permission.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Admin/GetProgramFile/{application}/{fileName}/{fileExtension}`

The URI for this method includes the following path elements:

Table 1. Path elements for the GET GetProgramFile method

Name	Type	Required?	Description
<i>{application}</i>	String	Yes	The name of the application for which the file is to be returned.
<i>{fileName}</i>	String	Yes	The name of the file in the batches directory that is to be returned. The <i>{fileName}</i> must not have a dot such as verify.1 or a plus sign such as batch+profiler.
<i>{fileExtension}</i>	String	Yes	The file name extension of the file that is to be returned, such as xml. Do not include the file name path, such as verify.xml.

Request content

The request for this method contains no content.

Response content

The response content is a byte stream.

This method returns one of the following response codes.

Table 2. Response codes for the GET GetProgramFile method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Admin/SetUserPermissionList

This method saves the job and task index pair list for which the user has permission.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Admin/SetUserPermissionList/{application}/{userInd}`

The URI for this method includes the following path elements.

Table 1. Path elements for the POST SetUserPermissionList method

Name	Type	Required?	Description
<code>{application}</code>	String	Yes	The name of the application for which the permissions list is to be saved.
<code>{userInd}</code>	String	Yes	The index for a user is the value in the <code>us_ind</code> column of the <code>tmuser</code> table in the Administration database.

Request content

The request content contains the job and task index pair list for which the user has permission.

Table 2. Request elements for the POST SetUserPermissionList method

Name	Description
Job	The value of <code>jb_ind</code> from the job table of the Administration database.
Task	The value of <code>ts_ind</code> from the tasks table of the Administration database.

Response content

The response for this method contains no content.

This method returns one of the following response codes.

Table 3. Response codes for the POST SetUserPermissionList method

Response Code	Description
---------------	-------------

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Admin/SetGroupPermissionList

This method saves the job and task index pair list for which the group has permission.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Admin/SetGroupPermissionList/{application}/{groupInd}`

The URI for this method includes the following path elements:

Table 1. Path elements for the POST method

Name	Type	Required?	Description
<code>{application}</code>	String	Yes	The name of the application for which the permissions list is to be saved.
<code>{groupInd}</code>	String	Yes	The index for a group is the value in the <code>gr_ind</code> column of the <code>tmgroup</code> table in the Admin database.

Request content

The request content contains the job and task index pair list for which the group has permission.

Table 2. Request elements for the POST SetGroupPermissionList method

Name	Description
Job	The value of <code>jb_ind</code> from the job table of the Administration database.
Task	The value of <code>ts_ind</code> from the tasks table of the Administration database.

Response content

The response for this method contains no content.

This method returns one of the following response codes.

Table 3. Response codes for the POST SetGroupPermissionList method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

POST SetGroupPermissionList method JSON request example

```

{"Permissions":[{"Job":"10","Task":"12"}]}
{"Permissions":[{"Job":"10","Task":"14"}]}
{"Permissions":[{"Job":"10","Task":"15"}]}
{"Permissions":[{"Job":"10","Task":"16"}]}

```

POST SetGroupPermissionList method XML request examples

```

<PermissionList><Permissions>
  <PermissionList.JobTaskPair><Job>10</Job><Task>12</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>10</Job><Task>14</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>10</Job><Task>15</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>10</Job><Task>16</Task>
</PermissionList.JobTaskPair>
</Permissions></PermissionList>

```

Parent topic: [Datacap Web Services REST API methods](#)

Admin/GetUserPermissionList

This method returns a job and task index pair list for which the user has permissions.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Admin/GetUserPermissionList/{application}/{userInd}`

The URI for this method includes the following path elements.

Table 1. Path elements for the GET GetUserPermissionList method

Name	Type	Required?	Description
<code>{application}</code>	String	Yes	The name of the application for which the permissions list is to be returned.

Name	Type	Required?	Description
{userInd}	String	Yes	The index for a user is the value in the us_ind column of the tmuser table in the Administration database.

Request content

The request for this method contains no content.

Response content

The response contains the job and task index pair list for which the user has permission. If -1 is returned for the task and job, the object has permission to all job and task pairs.

Table 2. Response elements for the GET GetUserPermissionList method

Name	Description
Job	The value of jb_ind from the job table of the Admin database.
Task	The value of ts_ind from the tasks table of the Admin database.

This method returns one of the following response codes.

Table 3. Response codes for the GET GetUserPermissionList method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request already included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

GET GetUserPermissionList method JSON response example

```
{ "Permissions": [{ "Job": "10", "Task": "15" }] }
{ "Permissions": [{ "Job": "12", "Task": "15" }] }
{ "Permissions": [{ "Job": "13", "Task": "15" }] }
{ "Permissions": [{ "Job": "14", "Task": "15" }] }
{ "Permissions": [{ "Job": "15", "Task": "15" }] }
{ "Permissions": [{ "Job": "16", "Task": "15" }] }
{ "Permissions": [{ "Job": "17", "Task": "15" }] }
{ "Permissions": [{ "Job": "18", "Task": "15" }] }
{ "Permissions": [{ "Job": "19", "Task": "15" }] }
{ "Permissions": [{ "Job": "20", "Task": "15" }] }
```

GET GetUserPermissionList method XML response example

```

<PermissionList><Permissions>
  <PermissionList.JobTaskPair><Job>10</Job><Task>15</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>12</Job><Task>15</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>13</Job><Task>15</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>14</Job><Task>15</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>15</Job><Task>15</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>16</Job><Task>15</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>17</Job><Task>15</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>18</Job><Task>15</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>19</Job><Task>15</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>20</Job><Task>15</Task>
</PermissionList.JobTaskPair>
</Permissions></PermissionList>

```

Parent topic: [Datacap Web Services REST API methods](#)

Admin/GetGroupPermissionList

This method returns a job and task index pair list for which the group has permissions.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Admin/GetGroupPermissionList/{application}/{groupInd}`

The URI for this method includes the following path elements:

Table 1. Path elements for the GET GetGroupPermissionList method

Name	Type	Required?	Description
<i>{application}</i>	String	Yes	The name of the application for which the permissions list is to be returned.
<i>{groupInd}</i>	String	Yes	The index for a group is the value in the gr_ind column of the tmgroup table in the Admin database.

Request content

The request for this method contains no content.

Response content

The response contains the job and task index pair list for which the group has permission. If -1 is returned for the task and job, the object has permission to all job and task pairs.

Table 2. Response elements for the GET GetGroupPermissionList method

Name	Description
------	-------------

Name	Description
Job	The value of <code>jb_ind</code> from the job table of the Admin database.
Task	The value of <code>ts_ind</code> from the tasks table of the Admin database.

This method also returns one of the following response codes.

Table 3. Response codes for the GET GetGroupPermissionList method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

GET GetGroupPermissionList method JSON response example

```
{ "Permissions": [{ "Job": "10", "Task": "12" }] }
{ "Permissions": [{ "Job": "10", "Task": "14" }] }
{ "Permissions": [{ "Job": "10", "Task": "15" }] }
{ "Permissions": [{ "Job": "10", "Task": "16" }] }
```

GET GetGroupPermissionList method XML response example

```
<PermissionList><Permissions>
  <PermissionList.JobTaskPair><Job>10</Job><Task>12</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>10</Job><Task>14</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>10</Job><Task>15</Task>
</PermissionList.JobTaskPair>
  <PermissionList.JobTaskPair><Job>10</Job><Task>16</Task>
</PermissionList.JobTaskPair>
</Permissions></PermissionList>
```

Parent topic: [Datacap Web Services REST API methods](#)

Admin/SaveTask

This method saves task properties such as the application name, description of the task, and program file name. You can queue the batch by a user or station and set the task properties mode for normal processing, batch creation, or task routing.

URI

`http://{IP address}:{Port}/ServiceWtm.svc/Admin/SaveTask`

Request content

The request contains the name of the application that is defined in the Application Manager and the following task properties content.

Table 1. Task properties for the POST method

Task Property	Description
conditions	A comma-separated list of conditions, such as <code>Skip_Verify</code> .
desc	The description of the task, such as Identify, Recognize, or Flag Problems.
id	The value of <code>ts_ind</code> from the tasks table of the Administration database. When you are creating a new task, <code>id</code> must be empty.
jobId	The job index value of <code>jb_ind</code> from the job table of the Administration database. The <code>jobId</code> is optional for an existing task but is required for a new task.
mode	Sets the task as normal processing, batch creation, or task routing. <ul style="list-style-type: none">• 2 - Normal processing• 3 - Batch creation• 6 - Routing task
name	The task name, such as <code>Profiler</code> .
program	The executable file or web page file that is used by the task, such as <code>FastDoc.exe</code> . You can use <code>Rulerunner.exe</code> for a background task when the tasks are run by a client.
queueBy	The value specifies the user and station to which the batch is queued. <ul style="list-style-type: none">• 0 - No user or station is specified• 1 - Station• 2 - User• 3 - Other Station• 4 - Other User• 5 - Station And User• 6 - Station And Other User• 7 - User And Other Station• 8 - Other Station And Other User
store	The value specifies the user ID or station ID that is stored with the batch. <ul style="list-style-type: none">• 0 - No user ID or station ID is stored.• 1 - The station ID is stored.• 2 - The user ID is stored.• 3 - The user ID and the station ID are stored.

Response content

The response for this method contains no content.

This method returns one of the following response codes.

Table 2. Response codes for the POST SaveTask method

Response Code	Description
---------------	-------------

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

POST SaveTask method JSON request example

```
{
  "application": "APT",
  "conditions": "Skip_Verify",
  "desc": "Identify, Recognize, Flag Problems",
  "id": "14",
  "jobId": "10",
  "mode": "3",
  "name": "Profiler",
  "program": "FastDoc.exe",
  "queueBy": "0",
  "store": "0"
}
```

POST SaveTask method XML request example

```
<TaskProperties>
  <application>APT</application>
  <conditions>Skip_Verify</conditions>
  <desc>Identify, Recognize, Flag Problems</desc>
  <id>14</id>
  <jobId>10</jobId>
  <mode>3</mode>
  <name>Profiler</name>
  <program>FastDoc.exe</program>
  <queueBy>0</queueBy>
  <store>0</store>
</TaskProperties>
```

Parent topic: [Datacap Web Services REST API methods](#)

Admin/GetMobileProfiles

This method returns the mobile profiles of the application for which you have permission. Each mobile profile represents one job that is enabled for mobile capture and includes image requirements and batch-level fields that can be entered on the mobile device.

The profile settings are a combination of task settings, batch fields, and batch level dictionaries in the Setup DCO. A list of the Start Batch Panel fields is saved to the profile on the mobile client.

In Datacap Web Client, you can set a batch creation task for a job to use the Mobile Capture program. The GET method adds a mobile profile to the returned list for each job and task pair. In addition to the task settings, the profile includes the job and task IDs, the job and task indexes, and batch fields. The batch fields include the *DataType* variables and a database column from the tmbatch table that is updated with the field value.

URI

http://{IP address}:{Port}/ServiceWM.svc/Admin/GetMobileProfiles/{application}

The URI for this method includes the following path elements.

Table 1. Path element for the GET GetMobileProfiles method

Name	Type	Required?	Description
{application}	String	Yes	The name of the application for which the mobile profiles are to be returned.

Request content

The request for this method contains no content.

Response content

The response for this method contains the mobile profiles for the Datacap application.

Table 2. Mobile Capture profile settings descriptions

Name	Description
<i>DataType</i>	The following are the values for the <i>DataType</i> node for the batch level fields. <ul style="list-style-type: none"> • 0 - None • 1 - Alphabetic A-Z • 2 - Alphanumeric A-Z, 0-9 • 4 - INT 16K, 32K plus or minus • 5 - INT 32 BigInteger • 6 - 32-bit floating point • 7 - 64-bit floating point • 8 - Date • 9 - Time HH:MM[:SS] • 16 - Currency
<i>DatabaseColumn</i>	The <i>DatabaseColumn</i> node for the batch level fields is not currently enabled.
<i>PageTypes Key and Value pair</i>	If the Setup DCO contains a dictionary that is called <i>PageNames</i> , the dictionary is returned. The <i>Key</i> is the DCO type and the <i>Value</i> is the display name. If the Setup DCO does not contain a dictionary, all page types are returned.
<i>DocumentTypes Key and Value pair</i>	If the Setup DCO contains a dictionary that is called <i>DocumentNames</i> , the dictionary is returned. The <i>Key</i> is the DCO type and the <i>Value</i> is the display name. If the Setup DCO does not contain a dictionary, all document types are returned.
<i>Job</i>	The job ID assigned to the job by Datacap.
<i>JobIndex</i>	The value of <i>jb_ind</i> from the job table of the Administration database.

Name	Description
<i>Task</i>	The task ID assigned to the task by Datacap.
<i>TaskIndex</i>	The value of ts_ind from the tasks table of the Administration database.

This method also returns one of the following response codes.

Table 3. The wTM response to the GET GetMobileProfiles method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

GET GetMobileProfiles method JSON response example

```
{
  "Count":2147483647,"Profiles":
  [{"BatchFields":
    {"Fields":
      [{"DataType":2147483647,
        "DatabaseColumn":"String content",
        "Name":"String content"}] },
    "DocumentTypes":
      {"Documents":
        [{"Key":"String content",
          "Value":"String content"}] },
    "Job":"String content",
    "JobIndex":2147483647,
    "PageTypes":
      {"Pages":
        [{"Key":"String content",
          "Value":"String content"}] },
    "Task":"String content",
    "TaskIndex":2147483647,
    "TaskSettings":
      {"Settings":
        [{"Name":"String content",
          "Settings":
            {"Settings":
              [{"Name":"String content",
                "Settings":{"Settings":null},
                "Value":"String content"}] },
          "Value":"String content"}] }
    }
  ]
}
```

GET GetMobileProfiles method XML response example

```

<MobileProfiles><Count>2147483647</Count>
  <Profiles>
    <MobileProfile>
      <BatchFields>
        <Fields>
          <BatchField>
            <DataType>2147483647</DataType>
            <DatabaseColumn>String content</DatabaseColumn>
            <Name>String content</Name>
          </BatchField>
        </Fields>
      </BatchFields>
      <DocumentTypes> <Documents>
        <Key>String content</Key>
        <Value>String content</Value>
      </Documents> </DocumentTypes>
      <Job>String content</Job>
      <JobIndex>2147483647</JobIndex>
      <PageTypes> <Pages>
        <Key>String content</Key>
        <Value>String content</Value>
      </Pages> </PageTypes>
      <Task>String content</Task>
      <TaskIndex>2147483647</TaskIndex>
      <TaskSettings>
        <Settings> <TaskSetting>
          <Name>String content</Name>
          <Settings>
            <Settings> <TaskSetting>
              <Name>String content</Name>
              <Settings> <Settings i:nil="true"> </Settings>
              <Value>String content</Value>
            </TaskSetting> </Settings>
          </Settings>
          <Value>String content</Value>
        </TaskSetting> </Settings>
      </TaskSettings>
    </MobileProfile>
  </Profiles>
</MobileProfiles>

```

Parent topic: [Datacap Web Services REST API methods](#)

Transaction/Start

This method starts a transaction and creates a new transaction ID. With the transaction endpoints, you can run rules without connecting to a Datacap Server, or database.

The response header includes a `wTmId` cookie and value that must be included in subsequent requests to maintain the session.

If an existing `wTmId` cookie is not provided and security is disabled in the configuration, a new `wTmId` will be returned in the `wTmId` cookie. The `wTmId` cookie must be included in all subsequent calls.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Transaction/Start`

Request content

The request for this method contains no content.

Response content

The response for this method returns the following attributes.

Table 1. Response content attributes for the Start GET method

Attribute name	Description of value
<code>{transactionId}</code>	The transaction ID that is assigned to the transaction by Datacap.

GET Transaction/Start method JSON response example

```
"String content"
```

GET Transaction/Start method XML response example

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">String  
content</string>
```

Parent topic: [Datacap Web Services REST API methods](#)

Transaction/SetFile

This method uploads a file to a transaction by using the transaction ID, file name, and file extension. A page object is not added to the page file.

To run rules, the following resource must be uploaded: a page file, such as scan.xml, containing references to page objects, if there are any.

If files are referenced in the page file and will be used by rules, those files must also be uploaded by using SetFile.

This method can be used to update a page file, for example, verify.xml. Files can be uploaded by streaming the file or by using the multipart/form-data content type. If you are streaming the file, specify the content type as the file type of the file that is being uploaded.

Files can be transferred as a binary string or by using Base64 encoding. When Base64 encoding is used, you must add `Content-Transfer-Encoding: base64` to the header when you are streaming, or as the last property in the message part before the file if you are using multipart/form-data. The SetFile request can accept a byte stream as the message body or can have the message body use the multipart/form-data format where the file byte stream is contained in each part.

If you specify a header field file name, then it is used as the file name for the uploaded file. If you do not specify a header field file name, then the `{fileName}` in the SetFile URI is used as the file name for the uploaded file.

You can adjust the value of the `maxAllowedContentLength` and `maxRequestLength` settings in the web.config file. The maximum size message that can be uploaded depends on your web server, proxy server, and client.

URI

```
http://{IP address}:{Port}/ServicewTM.svc/Transaction/SetFile/{transactionID}/{fileName}/{fileExtension}
```

The URI for this method includes the following path elements.

Table 1. Path elements for the POST SetFile method

Name	Type	Required?	Description
<i>{transactionId}</i>	String	Yes	The transaction ID for which the file is to be uploaded.
<i>{fileName}</i>	String	Yes	The name of the file that is to be uploaded. Note: The <i>{fileName}</i> must not have a dot such as verify.1 or a plus sign such as batch+profiler.
<i>{fileExtension}</i>	String	Yes	The file name extension of the file that is to be returned, such as exe. Do not include the file name path, set.xml.

Request content

The request content contains file that is to be uploaded.

Response content

The response for this method contains no content.

This method returns one of the following response codes.

Table 2. Response codes for the POST SetFile method

Response Code	Description
201 Created	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Transaction/Execute

This method runs rules for a transaction.

Rulesets is a comma separated list of rulesets to run in the order provided.

- *Workflow* - The workflow that is the value of the top level of a group of jobs and can be found in the web client administrator.
- *Messages* - A list of messages can be returned that are set by using a custom action.

URI

http://{IP address}:{Port}/ServiceWtm.svc/Transaction/Execute

Request content

The request content for this method runs the rules on a transaction.

Response content

The response for this method contains the same content as the Request content.

This method returns one of the following response codes.

Table 1. Response codes for the POST Execute method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

POST Transaction.Execute method XML request example

```
<Properties>
  <Application>String content</Application>
  <Id>String content</Id>
  <PageFile>String content</PageFile>
  <Rulesets>
    <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">String
content</string>
    <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">String
content</string>
  </Rulesets>
  <Workflow>String content</Workflow>
</Properties>
```

POST Transaction.Execute method JSON request example

```
{
  "Application": "String content",
  "TransactionId": "String content",
  "PageFile": "String content",
  "Rulesets": ["String content"],
  "Workflow": "String content"
}
```

POST Transaction.Execute method XML response example

```
<Properties>
  <DocumentCount>2147483647</DocumentCount>
  <Messages>
    <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">String
content</string>
    <string
xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">String
content</string>
  </Messages>
  <PageCount>2147483647</PageCount>
  <Status>2147483647</Status>
</Properties>
```

POST Transaction.Execute method JSON request example

```
{
  "DocumentCount":2147483647,
  "Messages":["String content"],
  "PageCount":2147483647,
  "Status":2147483647
}
```

Parent topic: [Datacap Web Services REST API methods](#)

Transaction/GetFile

This method returns a file in a transaction by using the transaction ID, file name, and file extension.

This method can be used to retrieve the recognition results in the data file (for example, tm000001.xml) or in the HTML output (for example, tm000001.htm).

URI

`http://{IP address}:{Port}/ServicewTM.svc/Transaction/GetFile/{transactionId}/{fileName}/{fileExtension}`

The URI for this method includes the following path elements.

Table 1. Path elements for the GET Transaction.GetFile method

Name	Type	Required?	Description
<code>{transactionId}</code>	String	Yes	The transaction ID for which the file is to be returned.
<code>{fileName}</code>	String	Yes	The name of the file in the file directory that is to be returned.
<code>{fileExtension}</code>	String	Yes	The file name extension of the file that is to be returned, such as xml. Do not include the file name path, such as verify.xml.

Request body

The request for this method contains no content.

Response content

The response returns the file. The response body is a byte stream.

This method returns one of the following response codes.

Table 2. Response codes for the GET Transaction.GetFile method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Transaction/GetFileList

This method gets a list of files in a transaction by using the transaction ID and mode.

When a file is uploaded, the created timestamp is set to the current time. When `GetFileList` is called, any files that were created by rules will be added to the list and `IsNew` will be true. Any files that were updated will have the modified property updated and `IsModified` will be true.

URI

`http://{IP address}:{Port}/ServicewTM.svc/Transaction/GetFileList/{transactionId}/{mode}`

The URI for this method includes the following path elements.

Table 1. Path elements for the GET GetFileList method

Name	Type	Required?	Description
<code>{transactionId}</code>	String	Yes	The transaction ID for which the list of files is to be returned.
<code>{mode}</code>	String	Yes	The { All, New, Modified } Mode values are All, New, Modified. New If the mode value is New, only files that were created by rules will be added to the list. Modified If the mode value is Modified, only files that were modified will be added to the list. .

Request content

The request for this method contains no content.

Response content

The response for this method returns a list of files.

This method also returns one of the following response codes:

Table 2. Response codes to the GET GetFileList method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service due to malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

GET GetFileList method JSON response example

```
[{
  "Created": "\/Date (928164000000-0400) \/",
  "IsModified": true,
  "IsNew": true,
  "Modified": "\/Date (928164000000-0400) \/",
  "Name": "String content"
}]
```

GET GetFileList method XML response example

```
<Files>
  <File>
    <Created>1999-05-31T11:20:00</Created>
    <IsModified>true</IsModified>
    <IsNew>true</IsNew>
    <Modified>1999-05-31T11:20:00</Modified>
    <Name>String content</Name>
  </File>
  <File>
    <Created>1999-05-31T11:20:00</Created>
    <IsModified>true</IsModified>
    <IsNew>true</IsNew>
    <Modified>1999-05-31T11:20:00</Modified>
    <Name>String content</Name>
  </File>
</Files>
```

Parent topic: [Datacap Web Services REST API methods](#)

Transaction/End

This method ends the transaction and removes files that are associated with the transaction ID.

URI

http://{IP address}:{Port}/ServiceWTM.svc/Transaction/End/{transactionId}

The URI for this method includes the following path elements.

Table 1. Path elements for the POST DeleteBatches method

Name	Type	Required?	Description
{transactionId}	String	Yes	The name of the transaction for which batches are to be deleted.

Request content

The request deletes the files that are associated with the transaction ID.

Response content

The response for this method contains no content.

This method returns one of the following response codes.

Table 2. Response codes for the POST DeleteBatches method

Response Code	Description
200 OK	The request was successful.
400 Bad Request	The request was not understood by the service because of malformed syntax.
401 Unauthorized	The request requires user authentication. If the request included valid credentials, the 401 response indicates that authorization was refused for those credentials.
405 Method Not Allowed	The method that is specified in the <code>Request-Line</code> is not allowed for the resource that is identified by the <code>Request-URI</code> . The response must include an <code>Allow</code> header that contains a list of valid methods for the requested resource.
411 Length Required	The service refuses to accept the request without a defined <code>Content-Length</code> . The client can repeat the request if it adds a valid <code>Content-Length</code> header field that contains the length of the message body in the request message.
500 Internal Error	An unexpected condition was encountered that prevented the service from fulfilling the request.

Parent topic: [Datacap Web Services REST API methods](#)

Fingerprint Maintenance Tool reference

Use the Fingerprint Maintenance Tool (FMT) to manage your fingerprints and synchronize information between the Fingerprint database, the Document Hierarchy, and Fingerprint XML data files.

- [Fingerprint Maintenance Tool configuration settings](#)
The Settings.ini file that is installed with the APT application in the C:\Datacap\APT\dco_APT folder contains information that is required by the Fingerprint Maintenance Tool.
- [Fingerprint Maintenance Tool buttons](#)
Use the following Fingerprint Maintenance Tool buttons to view, modify, and search for fingerprints.
- [Fingerprint Maintenance Tool fields](#)
List and description of the fields that you use for maintaining fingerprints. Fingerprints are used both for

page identification and for specifying recognition zones.

Related tasks:

[Maintaining fingerprints by using the Fingerprint Maintenance Tool](#)

Fingerprint Maintenance Tool configuration settings

The Settings.ini file that is installed with the APT application in the C:\Datacap\APT\dco_APT folder contains information that is required by the Fingerprint Maintenance Tool.

Database Section

Contains the connection strings to one or more databases that are used by the Fingerprint Maintenance Tool.

FingerprintDatabase Setting

Defines the connection string to the Fingerprint database used by the Fingerprint Maintenance Tool. For example:

```
FingerprintDatabase=Provider=Microsoft.Jet.OLEDB.4.0;  
Data Source=C:\Datacap\apt\process\APTFingerprint.mdb;  
Persist Security Info=False
```

Paths Section

Contains path information that is required by the Fingerprint Maintenance Tool.

FingerprintDirectory Setting

Defines the path to the application's folder that contains the application's fingerprint files (TIFF, CCO, or Fingerprint XML files). For example:

```
FingerprintDirectory=c:\datacap\apt\Fingerprint
```

FingerprintBackupDirectory Setting

Defines the path to the folder that contains backup copies of the Fingerprint XML files that are made when fingerprints are added to an application. If this folder does not exist, it is created when the Fingerprint Maintenance executable is started. For example:

```
FingerprintBackupDirectory=c:\datacap\apt\Fingerprint Backup
```

SetupDCO Setting

Defines the path to the application's Setup DCO file. For example:

```
SetupDCO=c:\datacap\apt\dco_APT\APT.xml
```

FMT Section

Contains the SQL query that is used by the Fingerprint Maintenance Tool. For example:

```
[FMT]  
FilteredSummary=Select Template.tp_TemplateID,Template.tp_DateAdded,  
Template.tp_HitCount,Template.tp_LastHit,Host.hs_RefName FROM Template,  
Host WHERE host.hs_HostID = Template.tp_HostID
```

Parent topic: [Fingerprint Maintenance Tool reference](#)

Fingerprint Maintenance Tool buttons

Use the following Fingerprint Maintenance Tool buttons to view, modify, and search for fingerprints.

Add Selected

Adds the selected fingerprints to the application's Fingerprint database and setup DCO, and copies the associated TIFF and CCO files into the application's Fingerprint folder.

Delete Selected

Deletes the selected fingerprint, including the TIFF, CCO, and fingerprint XML files, and the associated entry in the Fingerprint database.

Export Selected

Exports copies of the selected fingerprints, including the TIFF, CCO, and fingerprint XML files, to the application's \fingerprint\Fingerprint Export folder.

Find FPXML Files

Finds any fingerprint XML files that are in the application's \fingerprint folder.

Find Problems

Displays all fingerprints for which one or more components are missing from the Fingerprint Directory, such as the TIFF, CCO, or fingerprint XML files, or when the entry for the fingerprint in the application's Fingerprint database is missing.

Select All

Selects all displayed fingerprints.

Show Statistics

Displays all existing fingerprints and statistics that are related to each.

Parent topic: [Fingerprint Maintenance Tool reference](#)

Fingerprint Maintenance Tool fields

List and description of the fields that you use for maintaining fingerprints. Fingerprints are used both for page identification and for specifying recognition zones.

Backup Directory

Displays the full path to the folder that contains backup copies of the fingerprint XML files that are made when fingerprints are added to an application. If this folder does not exist, it is created when the Fingerprint Maintenance executable is started.

CCO

An X in this column indicates that the CCO file associated with this fingerprint is missing from the application's Fingerprint Directory.

Database

An X in this column indicates that the entry that is associated with this fingerprint is missing from the application's Fingerprint database.

Date Added

Displays the date that the fingerprint was either created in or added to this application from another.

Document Hierarchy

Displays the full path and file name of the application's Setup DCO.

Fingerprint

Displays the fingerprint's Fingerprint ID.

Fingerprint Database

Displays the connection string information that is related to the application's Fingerprint database.

Fingerprint Directory

Displays the full path to the directory in which the fingerprint-related files (TIFF, CCO, XML) are stored.

FPXML

An X in this column indicates that the XML file associated with this fingerprint is missing from the Fingerprint Directory.

Hits/Month

This field is an estimate of how many times per month the fingerprint is matched. This estimate is based on the age of the fingerprint and the number of times the fingerprint was matched to date. This estimate is calculated by first calculating the age of the fingerprint (in months) by the number of times it was matched.

For example, if a fingerprint is one month old (30 days) and was matched one time, the Hits/Month value is 1.0000. If the fingerprint is two months old (60 days) and was matched one time, the Hits/Month value is 0.5000.

HostName

The HostName is the Document Type assigned to the document associated with this fingerprint.

Last Hit

Displays the date on which the fingerprint was last matched.

Setup DCO

An X indicates that the Setup DCO is missing.

Status

The Status field indicates the status of the Fingerprint Maintenance Tool. The status is listed as Done, missing (denoted with an X), or Testing.

TIFF

An X in this column indicates the TIF file that is associated with this fingerprint is missing from the Fingerprint Directory.

Total Hits

Indicates the number of times the fingerprint was matched since it was created.

Parent topic: [Fingerprint Maintenance Tool reference](#)

Smart Parameter Special Variable Reference

Special variables are for use with smart parameters. Smart parameters do not work with all actions.

Check the action help  in Datacap Studio for compatibility information.

- [Special variables for accessing the application configuration file](#)
These variable return information that is related to the application configuration (.app) file.
- [Special variables for accessing the runtime hierarchy](#)
These variables return information that is related to documents, batches, pages, and fields.
- [Special variables for accessing job and task information](#)
These variables return metadata that is related to a particular job or task, such as the job name or operator.
- [Accessing Parent, Sibling and Child Objects with Smart Parameter Navigation](#)
When running rules in a Datacap application, you can use smart parameters to access information from parent or child DCO nodes.
- [Miscellaneous special variables](#)
Miscellaneous special variables can be used with Smart parameters.
- [Smart Parameter Build control](#)
In Datacap Studio, you can use the Smart Parameter Build control to create an expression that is evaluated at run time that allows you to obtain data from Datacap objects and variables. In Datacap Studio, when you select an action that takes a string parameter, the Parameters section displays the details of the selected action along with a button that displays three dots. By using this button, you can open Smart Parameter Build control.

Special variables for accessing the application configuration file

These variable return information that is related to the application configuration (.app) file.

- @APPPATH(<key_path>)
- @APPVAR(<key_path>)

Parent topic: [Smart Parameter Special Variable Reference](#)

@APPPATH(<key_path>)

Description

Retrieves the path to a file or folder from the application's configuration (.app) file. Use the Datacap Application Manager to modify the information that is contained in this file. Do not edit the file directly.

Syntax

```
@APPPATH(key_path)
```

Arguments

key_path Specifies the path through the XML hierarchy to field you want:

- If the field name is unique within the file, you can specify just the field name.
- If the field name is not unique within the file, you must specify the path to the instance you want. For example, you might have multiple workflows. If you do not specify the path, you get the first instance. To obtain the path, move the mouse pointer over the field in the Datacap Application Manager and read the path from the tooltip.

Examples

You can use @APPPATH to retrieve the following information from the application configuration file.

Field name	Key name	Example	Notes
Application fields			
Batches folder	runtime	@APPPATH(runtime)	(Field name is unique)
Export folder	export	@APPPATH(export)	(Field name is unique)
Fingerprint folder	fingerprint	@APPPATH(fingerprint)	(Field name is unique)
Workflow fields			
Setup DCO	setupdco	@APPPATH(setupdco)	(assumes single workflow)
Rules folder	rules	@APPPATH(dco_Workflow1/rules)	(specifies Workflow 1)
VScan source folder	vscanimagedir	@APPPATH(vscanimagedir)	(assumes single workflow)

Field name	Key name	Example	Notes
Imagefix INI	imagefix	@APPPATH(dco_Workflow2/imagefix)	(specifies Workflow 2)

Parent topic: [Special variables for accessing the application configuration file](#)

@APPVAR(<key_path>)

Description

Retrieves a connection string, value, or other attribute from the application configuration (.app) file. Use the Datacap Application Manager to modify the information that is contained in this file. Do not edit the file directly.

Attention: Before Datacap 8.0, this parameter returns the value for the specified variable as defined in the [Variables] section in paths.ini. The .ini file is in the process folder of the application.

The Datacap Application Manager encrypts the connection strings and decryption is handled by the standard actions.

Syntax

```
@APPVAR(key_path[:attribute])
```

Arguments

key_path - Specifies the path to field you want. See @APPPATH for details.

attribute - Specifies the attribute name (optional):

- For custom values, use "v" (if you do not specify an attribute, "v" is assumed).
- For connection strings, use "cs".

The easiest way to get the path and attribute is to look in the Datacap Application Manager:

- On the Application and Datacap tabs, move the mouse pointer over the field and read the smart parameter value from the tooltip.
- On the Custom Values tab, read the smart parameter value that is displayed in each section.

Examples

You can use @APPVAR to retrieve the following information from the application configuration file.

Field name	Key name:attribute	Example	Notes®
Workflow fields			
Lookup database	lookupdb:cs	@APPVAR(lookupdb:cs)	(Workflow 1)
Fingerprint database	fingerprintconn:cs	@APPVAR(fingerprintconn:cs)	(one workflow)
Export database	exportdb:cs	@APPVAR(dco_Wkflw1/exportdb:cs)	(Workflow 1)

Field name	Key name:attribute	Example	Notes®
Datacap fields			
Engine database	tmengine:cs	@APPVAR(tmengine:cs)	(Field is unique)
Admin database	tadmin:cs	@APPVAR(tadmin:cs)	(Field is unique)
Custom values			
General string values	values/gen/<value_name>	@APPVAR(values/gen/Value1)	(Value1)
Connection strings	values/dsn/<value_name>:cs	@APPVAR(values/dsn/DB1:cs)	(CS for DB1)
Datacap connection strings	values/tmsdn/<value_name>:cs	@APPVAR(values/tmsdn/TMDB1:cs)	(CS for TMDB1)
Advanced values	values/adv/<value_name>	@APPVAR(values/adv/Value1)	(Value1)

Parent topic: [Special variables for accessing the application configuration file](#)

Related information:

[Storing passwords, connection strings, and other parameters in the .app file](#)

Special variables for accessing the runtime hierarchy

These variables return information that is related to documents, batches, pages, and fields.

- [@BATCHID](#)
- [@ID](#)
- [@STATUS](#)
- [@VALUE](#)
- [@VAR\(<variable_name>\)](#)
- [@B.<variable_name>](#)
- [@D.<variable_name>](#)
- [@P.<variable_name>](#)
- [@F.<variable_name>](#)
- [@X.<variable_name>](#)
- [@B\<field_name>\[.<variable_name>\]](#)
- [@D\<field_name>\[.<variable_name>\]](#)
- [@P\<field_name>\[.<variable_name>\]](#)
- [@F\<field_name>\[.<variable_name>\]](#)
- [@X\<field_name>\[.<variable_name>\]](#)

Parent topic: [Smart Parameter Special Variable Reference](#)

@BATCHID

Description

Returns the value of the id attribute for the current batch.

Example

In this example, the smart parameter returns the ID of the current batch.

Action and return value	XML example (if applicable)
Action: rr_Get("@BATCHID") Return value: 20110046.001	<B id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar002_20110046.001"> <V n="TYPE">TravelDocs</V> <V n="STATUS">1</V>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@ID

Description

Returns the value of the id attribute for the current object. For example, if the rule that contains this special variable is bound to a page, the current object is the current page.

Example

In this example, the rule that contains the action is bound to a page and the smart parameter returns the ID of the current page.

Action and return value	XML example (if applicable)
Action: rr_Get("@ID") Return value: TM000001	<P id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar015_TM000001"> <V n="TYPE">Rental_Agreement</V> <V n="STATUS">1</V>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@STATUS

Description

Returns the value of the STATUS variable for the current object. STATUS is also a setup property and might specify a special characteristic (for example, -1 indicates a hidden field).

Example

In this example, the rule that contains the action is bound to a page and the smart parameter returns the status of the current page.

Action and return value	XML example (if applicable)
Action: rr_Get("@STATUS") Return value: 1	<P id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvvar026_TM000001"> <V n="TYPE">Rental_Agreement</V> <V n="STATUS">1</V>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@VALUE

Description

Returns the text of the current object (usually a field).

Example

In this example, the rule that contains the action is bound to a field and the smart parameter returns the text of the current field. If the current object is not a field, @VALUE looks for a value in a variable *Text* of the calling object.

Action and return value	XML example (if applicable)
Action: rr_Get("@VALUE") Return value: SUV	<F id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvvar032_Car_Type"> <V n="TYPE">Car_Type</V> <C cn="10" cr="588,748,600,769">83</C> <-- ASCII `S` <C cn="10" cr="605,748,620,769">85</C> <-- ASCII `U` <C cn="10" cr="625,748,643,769">86</C> <-- ASCII `V` </F>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@VAR(<variable_name>)

Description

Returns the value of the specified variable on the current object.

Example

In this example, the rule that contains the action is bound to a page. The smart parameter returns the value of the *TYPE* variable for the current page.

Action and return value	XML example (if applicable)

Action and return value	XML example (if applicable)
Action: <code>rr_Get("@VAR(TYPE)")</code>) Return value: Rental_Agreement	<pre><<P id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.refere nce.doc_dcvar033_TM000001"> <V n="TYPE">Rental_Agreement</V> <V n="STATUS">1</V></pre>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@B.<variable_name>

Description

Accesses the value of the specified variable on batch DCO node. The current DCO node does not need to be batch level. The current node can be a batch, document, page or field node and this syntax accesses the parent batch node.

Example

In this example, the smart parameter returns the value of the *TYPE* variable for the batch. The second example creates a new variable *Name* at the batch level and sets the value to “Fred”.

Action and return value	XML example (if applicable)
Action: <code>rr_Get("@B.TYPE")</code> Return value: TravelDocs <code>rrSet("Fred","@B.Name")</code> Creates the variable <i>Name</i> and sets its value to “Fred”	<pre><B id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc .reference.doc_dcvar104_20110046.001"> <V n="TYPE">TravelDocs</V> <V n="STATUS">1</V></pre>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@D.<variable_name>

Description

Accesses the value of the specified variable on the current, or parent, document DCO node. The current DCO node does not need to be a document level node. The current node can be a document, page or field node and this syntax accesses the parent document node.

Example

In the first example, the smart parameter returns the value of the *TYPE* variable for the current document. In the second example, the *TYPE* is changed to *Hotel_Rental*.

Action and return value	XML example (if applicable)
-------------------------	-----------------------------

Action and return value	XML example (if applicable)
Action: <code>rr_Get("@D.TYPE")</code> Return value: Car_Rental <code>rrSet("Hotel_Rental", "@D.TYPE")</code> Sets the variable <i>TYPE</i> to <i>Hotel_Rental</i>	<pre><D id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar105_20110046.001.01"> <V n="TYPE">Car_Rental</V> <V n="STATUS">0</V></pre>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@P.<variable_name>

Description

Accesses the value of the specified variable on the current, or parent, page DCO node. The current DCO node does not need to be field level. The current node can be a page or field node and this syntax accesses the parent page node.

Example

In this example, the smart parameter returns the value of the *TemplateID* variable for the current page.

Action and return value	XML example (if applicable)
Action: <code>rr_Get("@P.TemplateID")</code> Return value: 556	<pre><P id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar024_TM000001"> <V n="TYPE">Rental Agreement</V> <V n="TemplateID">555</V></pre>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@F.<variable_name>

Description

Accesses the value of the specified variable within the current field.

Example

In this example, the smart parameter returns the value of the *TYPE* variable for the current field.

Action and return value	XML example (if applicable)

Action and return value	XML example (if applicable)
Action: <code>rr_Get("@F.TYPE")</code>) Return value: Pickup_Date	<pre><F id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference .doc_dcvar014_Pickup_Date"> <V n="TYPE">Pickup_Date</V> <V n="Position">183,402,535,463</V></pre>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@X.<variable_name>

Description

Accesses the text of the specified variable on the current node. This smart parameter enhances the portability of the rules by working generically on the current DCO node. For example, a function that is written for the page level can be reused at another level by simply tying the ruleset to a new DCO object without having to update the @X smart variable to the new level .

Example

In the first example, the smart parameter returns the value of the *TYPE* variable for the current document. In the second example, the *TYPE* is changed to *Hotel_Rental*.

Action and return value	XML example (if applicable)
-------------------------	-----------------------------

Action and return value	XML example (if applicable)
<p>Action:</p> <pre> RegexFind_InBlock("Invoice..") GoSiblingBlockNext("1") GetSelectedBlockType("@X.SelectedBlock") </pre> <p>Return value: The variable <i>SelectedBlock</i> is set to "Word". If the variable <i>SelectedBlock</i> does not exist, it is created.</p>	<pre> <Block pos="1242,794,1593,819" s="4"> <L pos="1242,794,1593,819" s="4"> <W pos="1242,794,1381,818" v="Invoice#:" s="4" cn="999999999"> <C pos="1242,794,1247,818" v="I" s="4" cn="9" /> <C pos="1251,800,1267,818" v="n" s="4" cn="9" /> <C pos="1270,801,1287,818" v="v" s="4" cn="9" /> <C pos="1289,800,1307,819" v="o" s="4" cn="9" /> <C pos="1311,794,1315,818" v="i" s="4" cn="9" /> <C pos="1319,800,1335,819" v="c" s="4" cn="9" /> <C pos="1337,800,1354,819" v="e" s="4" cn="9" /> <C pos="1355,794,1373,818" v="#" s="4" cn="9" /> <C pos="1377,801,1381,818" v=":" s="4" cn="9" /> </W> <W pos="1503,794,1593,818" v="23857" s="4" cn="99999"> <C pos="1503,794,1519,818" v="2" s="4" cn="9" /> <C pos="1522,794,1538,819" v="3" s="4" cn="9" /> <C pos="1540,794,1556,819" v="8" s="4" cn="9" /> <C pos="1559,795,1575,819" v="5" s="4" cn="9" /> <C pos="1577,795,1593,818" v="7" s="4" cn="9" /> </W> </L> </Block> </pre>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@B\`<field_name>`[.`<variable_name>`]

Description

Accesses the text of the specified field on the current batch, or the value of the specified variable of the specified field on the current batch.

Example

In this example, the smart parameter returns the value of the Text variable for the batch-level field *Field2*

Action and return value	XML example (if applicable)
-------------------------	-----------------------------

Action and return value	XML example (if applicable)
Action: <code>rr_Get("@B\Field2)</code> Return value: The text value of <i>Field2</i>	<pre> <B type="APT"> <V n="ID">0</V> <V n="TYPE">Batch</V> <V n="STATUS">0</V> <V n="MIN TYPES">0</V> <V n="MAX TYPES">0</V> <V n="rules">&lt;in /&gt;&lt;out /&gt; </V> <P type="Other" pos="0" min="0" max="0"/> <D type="Document" pos="0" min="0" max="0"/> <F type="Field2" pos="0" min="0" max="0"/> </pre>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@D<field_name>[.<variable_name>]

Description

Accesses the text of the specified field on the current or parent document, or the value of the specified variable of the specified field on the current document.

Example

In this example, the smart parameter returns the value of the *TYPE* variable for the field “Field1”, which is directly attached to the current document.

Action and return value	XML example (if applicable)
Action: <code>rr_Get("@D\Page\Field1.TYPE")</code> Return value: Field	<pre> <F type="Field1"> <V n="ID">0</V> <V n="TYPE">Field</V> <V n="STATUS">0</V> <V n="Position">0,0,0,0</V> <V n="MIN TYPES">0</V> <V n="MAX TYPES">0</V> <V n="length"></V> <V n="ReqConf">8</V> <V n="rules">&lt;in /&gt;&lt;out /&gt;</V> </F> </pre>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@P<field_name>[.<variable_name>]

Description

Finds the parent page of the calling DCO. Then, find the child with the *dco_id* or *dco_type* of the parent page. Then, find the variable of this name on the specified child of the parent page.

The character values used in the following examples are ASCII codes, but they might be Unicode as well.

Example

In the first example, the smart parameter returns the text of the Car_Type field on the current page.

In the second example, the smart parameter returns the value of the TYPE variable of the Car_Type field.

Action and return value	XML example (if applicable)
Action: rr_Get("@P\Car_Type") Return value: SUV	<pre><F id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar023_Car_Type"> <V n="TYPE">Car_Type</V> <C cn="10" cr="588,748,600,769">83</C> <-- ASCII `S' <C cn="10" cr="605,748,620,769">85</C> <-- ASCII `U' <C cn="10" cr="625,748,643,769">86</C> <-- ASCII `V' </F></pre>
Action: rr_Get("@P\Car_Type.TYPE") Return value: Car_Type	<pre><F id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar023_Car_Type"> <V n="TYPE">Car_Type</V> <C cn="10" cr="588,748,600,769">83</C> <-- ASCII `S' <C cn="10" cr="605,748,620,769">85</C> <-- ASCII `U' <C cn="10" cr="625,748,643,769">86</C> <-- ASCII `V' </F></pre>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@F\<field_name>[.<variable_name>]

Description

Accesses the text of the specified subfield of the current field, for example, a field within a line item. Or returns the value of the specified variable of the specified subfield.

Example

In these examples, the rule that contains the action is bound to a field with subfields. In the first example, the smart parameter returns the text of the Unit_Cost subfield of the current field.

In the second example, the smart parameter returns the value of the Unit_Cost subfield TYPE variable.

Action and return value	XML example (if applicable)
Action: rr_Get("@F\Unit_Cost") Return value: \$9.90	<pre><F id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar013_Other_Charges_Line_Item0"> <V n="TYPE">Other_Charges_Line_Item</V> <F id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar013_Unit_Cost"> <V n="TYPE">Unit_Cost</V> <C cn="10" cr="1290,511,1305,540">36</C> <-- ASCII ` \$ ' <C cn="10" cr="1308,515,1321,536">57</C> <-- ASCII ` 9 ' <C cn="10" cr="1325,533,1329,536">46</C> <-- ASCII ` . ' <C cn="10" cr="1334,515,1348,536">57</C> <-- ASCII ` 9 ' <C cn="10" cr="1350,515,1365,536">48</C> <-- ASCII ` 0 ' </F></pre>
Action: rr_Get("@F\Unit_Cost.TYPE") Return value: Unit_Cost	<pre><F id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar013_Unit_Cost"> <V n="TYPE">Unit_Cost</V> <C cn="10" cr="1290,511,1305,540">36</C> <-- ASCII ` \$ ' <C cn="10" cr="1308,515,1321,536">57</C> <-- ASCII ` 9 ' <C cn="10" cr="1325,533,1329,536">46</C> <-- ASCII ` . ' <C cn="10" cr="1334,515,1348,536">57</C> <-- ASCII ` 9 ' <C cn="10" cr="1350,515,1365,536">48</C> <-- ASCII ` 0 ' </F></pre>

Parent topic: [Special variables for accessing the runtime hierarchy](#)

@X\<field_name>[.<variable_name>]

Description

Accesses the text of the specified field on the current node, or the value of the specified variable of the specified field on the current node. This smart parameter enhances the portability of the rules by providing generic access to the current DCO node.

For example, a function that is written for the page level can be reused at another level by simply tying the ruleset to a new DCO object without having to update the `@X` smart parameter to the new level.

Example

This example shows actions being called on a document level node.

Action	XML example (if applicable)
Action: <code>rr_Get("@X.TYPE")</code> Return Value: Car_Rental	D id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.r eference.doc_dcvr103_20110046.001.01"> <V n="TYPE">Car_Rental</V> <V n="STATUS">0</V>
Action: <code>rrSet("Hotel_Rental", "@X.TYPE")</code> Sets the variable <code>TYPE</code> to "Hotel_Rental"	

Parent topic: [Special variables for accessing the runtime hierarchy](#)

Special variables for accessing job and task information

These variables return metadata that is related to a particular job or task, such as the job name or operator.

- [@JOBID](#)
- [@JOBNAME](#)
- [@OPERATOR](#)
- [@STATION](#)
- [@TASKID](#)
- [@TASKNAME](#)

Parent topic: [Smart Parameter Special Variable Reference](#)

@JOBID

Description

Returns the ID of the current job. This ID is the value that is specified in the ID field on the Datacap Administrator Workflow tab.

Example

In this example, the smart parameter returns the ID of the current job.

Action and return value	XML example (if applicable)
Action: rr_Get("@JOBID")	
Return value: Main Job	

Parent topic: [Special variables for accessing job and task information](#)

@JOBNAME

Description

Returns the name of the current job. This name is the value that is specified in the Description field on the Datacap Administrator Workflow tab.

Example

In this example, the smart parameter returns the name of the current job.

Action and return value	XML example (if applicable)
Action: rr_Get("@JOBNAME")	
Return value: Main Job	

Parent topic: [Special variables for accessing job and task information](#)

@OPERATOR

Description

Returns the user name of the person who ran the job.

Example

In this example, the smart parameter returns the name of the person who ran the job.

Action and return value	XML example (if applicable)
Action: rr_Get("@OPERATOR")	
Return value: admin	

Parent topic: [Special variables for accessing job and task information](#)

@STATION

Description

Returns the ID of the station that runs the job. This ID is the value that is specified in the ID field on the Datacap Administrator Workflow tab.

Example

In this example, the smart parameter returns the ID of the station that runs the current job.

Action and return value	XML example (if applicable)
Action: rr_Get("@STATION")	
Return value: 1	

Parent topic: [Special variables for accessing job and task information](#)

@TASKID

Description

Returns the ID of the current task. This ID is the value that is specified in the ID field on the Datacap Administrator Workflow tab.

Example

In this example, the smart parameter returns the ID of the current task.

Action and return value	XML example (if applicable)
Action: rr_Get("@TASKID")	
Return value: Export	

Parent topic: [Special variables for accessing job and task information](#)

@TASKNAME

Description

Returns the name of the current task. This name is the value that is specified in the Description field on the Datacap Administrator Workflow tab.

Example

In this example, the smart parameter returns the name of the current task.

Action and return value	XML example (if applicable)
Action: rr_Get("@TASKNAME")	
Return value: Export	

Parent topic: [Special variables for accessing job and task information](#)

Accessing Parent, Sibling and Child Objects with Smart Parameter Navigation

When running rules in a Datacap application, you can use smart parameters to access information from parent or child DCO nodes.

This feature allows the rules running on the current object to access other objects and their variables for both reading and writing. For example, while on a page level node, it is possible to access the parent batch or document node for the object, or navigation can be to a child field object.

Smart Parameter Navigation BNF Syntax

`<expression>: <level>[.<variable >] | <level>\<type>[\<type>][.<variable>]`

`<level>: @B | @D | @P | @F | @X`

`<variable>`: The name of a variable in the DCO.

`<type>`: The ID of a DCO object.

- [Level Identifier](#)
- [Variable Identifier](#)
- [Type Identifier](#)

Parent topic: [Smart Parameter Special Variable Reference](#)

Level Identifier

Description

The level identifiers, B, D, P, F, X denote the starting level of the expression. The level is always relative to the current DCO object. B = Batch level, D = Document level, P = Page level, F = Field level, X = current level.

For example, if the current DCO is a page object, then @P accesses the current object, as will @X.

@B accesses the parent batch object and @D accesses the parent document.

When a level identifier is used alone, without an identifying DCO name, then the object that is accessed is in the direct parent or child hierarchy of the current object.

As a review, the typical object hierarchy is: Batch -> Document -> Page -> Field.

A batch can have 1 or more documents, a document can have 1 or more pages and a page can have one or more fields. In some situations, a field can also have one or more child fields.

Parent topic: [Accessing Parent, Sibling and Child Objects with Smart Parameter Navigation](#)

Variable Identifier

Description

The variable identifier is the name of the variable to access on the specified DCO object. If a variable does not exist, the contents are considered to be empty. If a variable name is not specified, then the “TEXT” variable of the object is accessed, making @X equivalent to @X.TEXT.

Example

For example, assuming the current object is a page level object:

@X.MyVar accesses “MyVar” on the current object.

`@P.MyVar` also accesses “MyVar” on the current page.

`@D.MyVar` accesses “MyVar” on the parent document object.

`@F.MyVar` accesses “MyVar” on the current field object.

When a variable name is not specified, then the “TEXT” variable of the object is accessed. Again, using the example of these smart parameters on a page level object:

`@X` accesses “TEXT” on the current object.

`@P` also accesses “TEXT” on the current page.

`@D` accesses “TEXT” on the parent document object.

`@F` accesses the “TEXT” property on current field object.

Example of Variable Use In Actions

You can use Smart parameters and navigation as parameter input to any action whose parameters support smart parameters. The Smart Parameters provide an additional and powerful tool to create applications in which the input parameter value can be determined at runtime. The `rrSet`, `rrCompare`, and `rrCompareNot` are some of the common actions that use smart parameters, but the usage of Smart Parameters is not limited to these actions.

The following is a page level example where a barcode type is checked and if it is “Invoice”, the page type is set to “Invoice”.

```
rrCompare("@X. GetBarCode", "Invoice")
SetPageType("Invoice")
```

The following example shows to how accomplish the same objective in a different way, where the page type is always set to the barcode value directly.

```
SetPageType("@X. GetBarCode")
```

Note:

The application in the above examples must handle the cases where the page is an unexpected type or the barcode is not read.

In the following page level example, a FileNet property is set to the ID of the document level object.

```
FNP8_SetProperty("DocumentTitle", "@D.ID")
```

In the following field level example, a FileNet property is set to the text value of the field, or as in the second example the ‘TEXT’ variable value if the calling object is a Batch, Document or Page DCO node.

```
FNP8_SetProperty("InvoiceNumber", "@F") OR FNP8_SetProperty("InvoiceNumber", "@X")
```

Parent topic: [Accessing Parent, Sibling and Child Objects with Smart Parameter Navigation](#)

Type Identifier

Description

The name identifier in the smart parameter navigation syntax allows access to sibling nodes, either above or below the current object. The target object is identified by the object ID. The navigation expression must start

with the target level and the object ID.

To illustrate, consider the following hierarchy and types:

B T1 -->	D T2 -->	P T3 -->	F T4
			F T5
		P T6 -->	F T7
			F T8

This hierarchy always has only one batch object. This runtime DCO contains 2 document objects, 3 page objects and 6 field objects. The document object ID2 contains 2 pages, each with 2 fields. The “T” is a shorthand way to indicate the different object type names.

The following examples show navigation of the DCO using the hierarchy illustrated above.

If the current object is page T3, this accesses the TEXT value second field: “@F|T5”. Here, the variable “MyVar” is accessed on the same field object: “@F|T5.MyVar”.

The following example accesses the second field of the second page from the document level object T2:

```
@D\T6\T8.MyVar
```

If on the page object T3, this example accesses the sibling page T6:

```
@D\T6.MyVar
```

Example of Type Use In Actions

There are some cases where specifying navigation of specific types is useful. Here is an example where the current object is a document object which is creating a PDF of all pages and the author of the PDF is contained in a field object of the main page. The value of the `y_PDFAuthor` variable is saved to the calling document object for later use by a subsequent action; for example when creating a PDF.

```
rrSet("@P\main_page\myfield.TEXT", "@X.y_PDFAuthor")
```

Another similar example where a document level object sets a FileNet property obtained from a field object.

```
FNP8_SetProperty("DocumentTitle, @P\mypage\myfield.TEXT ")
```

Parent topic: [Accessing Parent, Sibling and Child Objects with Smart Parameter Navigation](#)

Miscellaneous special variables

Miscellaneous special variables can be used with Smart parameters.

- [@CHR\(<unicode_value>\)](#)
- [@DATE\(<format>\)](#)
- [@DCO\(<property_name>\)](#)
- [@DICT_VALUE\(<field>\)](#)
- [@DICT_WORD\(<field>\)](#)
- [@DICT_VINDEX\(<csv_string>\)](#)
- [@DICT_WINDEX\(csv_string\)](#)
- [@EMPTY](#)
- [@LOCALE](#)
- [@PATH\(<key>\)](#)

- @PILOT(<property_name>)
- @PROJECTDIR
- @PROCESSDIR
- @STRING(<string_value>)
- @TIME(<format>)
- @TYPE

Parent topic: [Smart Parameter Special Variable Reference](#)

@CHR(<unicode_value>)

Description

Returns the character corresponding to the specified UNICODE.

Example

In this example, the smart parameter returns the character corresponding to UNICODE value 38.

Action and return value	XML example (if applicable)
Action: rr_Get("@CHR(38)") Return value: &	

Parent topic: [Miscellaneous special variables](#)

@DATE(<format>)

Description

Returns the current date in the format specified. The default format is "MM/DD/YYYY".

To set a @DATE attribute, ensure that the date value is in one of the date formats accepted by IBM Content Manager.

Valid date formats are:

- "yyyy-MM-dd"
- "yyyy/MM/dd"
- "MM-dd-yyyy"
- "MM/dd/yyyy"

You can reformat a date field by calling the `Validations.IsFieldDateWithReformat` function on the current field.

```
IsFieldDateWithReformat ("yyyy/MM/dd")
```

Example

In this example, the smart parameter returns the current date.

Action and return value	XML example (if applicable)

Action and return value	XML example (if applicable)
Action: rr_Get("@DATE(mm.dd.yyyy)") Return value: 12.31.2010	

Parent topic: [Miscellaneous special variables](#)

@DCO(<property_name>)

Description

Returns the value of the specified DCO object property. The DCO object is an internal data structure that contains the current runtime batch hierarchy information. This information includes the batch ID (ID), batch type (TYPE), batch status (STATUS), and the full runtime batch hierarchy XML.

Example

In this example, the rule that contains the action is bound to a page. The example also contains the smart parameter returns the portion of the runtime batch hierarchy XML for the current page.

Action and return value	XML example (if applicable)
Action: rr_Get("@DCO(XML)") Return value: <?xml-stylesheet type="text/xsl" href="#_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_...\dco.xsl"?><P id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar006_TM000001"><F id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar006_Pickup_Date"> <V n="TYPE">Pickup_Date</V><V n="Position">0,0,0,0</V><V n="STATUS">0</V></F>	

Parent topic: [Miscellaneous special variables](#)

@DICT_VALUE(<field>)

Description

This special variable works with OMR fields (RecogType=4) that are bound to a dictionary. It returns the dictionary values corresponding to the items that are selected in the specified OMR field.

Example

In this example, the rule that contains the rr_Get action is bound to a page that contains an OMR field (Options). The OMR field has three subfields and is bound to the dictionary shown in the XML code. On the current page all three options are selected, so the return string contains all three dictionary values.

Action and return value	XML example (if applicable)
Action: rr_Get("@DICT_VALUE(Options)") Return value: Navigation System Child Seat Fuel Service	<pre><DICT n="Options"> <W v="Navigation System">Navigation System</W> <W v="Child Seat">Child Seat</W> <W v="Fuel Service">Fuel Service</W> </DICT></pre>

Parent topic: [Miscellaneous special variables](#)

@DICT_WORD(<field>)

Description

Same as @DICT_VALUE, except this special variable returns the dictionary words corresponding to the selected items.

Parent topic: [Miscellaneous special variables](#)

@DICT_VINDEX(<csv_string>)

Description

This special variable works with actions bound to an OMR field (RecogType=4) where the OMR field is bound to a dictionary. The parameter returns a string of 1's and 0's corresponding to the dictionary values you specify as a comma-separated list.

Example

In this example, the rule that contains the rr_Get action is bound to an OMR field. The OMR field has three subfields and is bound to the dictionary shown on the right. The values that are specified correspond to the second and third items in the dictionary, so the return value is 011. The rr_Get sets the character values on the three OMR subfields to 0, 1, and 1.

Action and return value	XML example (if applicable)
Action: rr_Get("@DICT_VINDEX(Fuel Service,Child Seat)") Return value: 011	<pre><DICT n="Options"> <W v="Nav System">Nav System</W> <W v="Child Seat">Child Seat</W> <W v="Fuel Service">Fuel Service</W> </DICT></pre>

Parent topic: [Miscellaneous special variables](#)

@DICT_WINDEX(csv_string)

Description

Same as @DICT_VINDEX, except the argument for this special variable uses the dictionary words.

Parent topic: [Miscellaneous special variables](#)

@EMPTY

Description

This special variable represents an empty string.

Example

In this example, rrSet clears the custom page variable *MyVar*.

Action and return value	XML example (if applicable)
Action: rrSet("@EMPTY","@P.MyVar")	

Parent topic: [Miscellaneous special variables](#)

@LOCALE

Description

Returns one of the following values:

hr_locale	The value of variable hr_locale is returned if the value of that variable is not empty. For information, see hr_locale .
System locale	Otherwise, if the value of hr_locale is empty, the system locale is returned.

Example

Action	XML example (if applicable)	Description
<code>rrSet("@LOCALE", "@X.CurrentLocale")</code>		Shows how to save the current locale so that you can later determine what locale was in effect (such as after batch processing). The current locale is saved in the following manner: @LOCALE obtains the current locale, and rrSet saves that value in a variable that is called CurrentLocale for the current DCO object.
<code>rrSet("@LOCALE", "@X.CurrentLocale") rrSet("en-GB", "@X.hr_locale") // perform actions that are // affected by // locale such as recognition or // validations rrSet("@X.CurrentLocale", "@X.hr_locale")</code>		Shows how to save and restore the original locale so that you can temporarily use a different locale.

Parent topic: [Miscellaneous special variables](#)

@PATH(<key>)

Description

Returns the full path for the specified identifier as defined in the [PATHS] section of paths.ini, in the dco_<app_name> folder of the application. The function that is supported by this special variable was replaced by the Datacap Application Manager and the @APPPATH special variable.

Example

In this example, the smart parameter returns the path to the image folder of the APT application. This folder is defined in the paths.ini file that is shown on the right.

Action and return value	XML example (if applicable)
Action: rr_Get("@PATH(VscanImageDir)") Return value: C:\Datacap\APT\Images\Input	[Paths] VscanImageDir=..\Images\Input ProcessDir=..\dco_APT RRXDir=..\..\dco_APT\Rules FingerprintDir=..\Fingerprint ExportDir=..\Export

Parent topic: [Miscellaneous special variables](#)

@PILOT(<property_name>)

Description

Returns the value of the specified Pilot object property. The Pilot object is an internal data structure that is configured by Datacap at the start of task execution. It contains the information that is required to run the task. Such as the batch folder (BATCHDIR), the batch ID (BATCHID), the input DCO file (DCOFILE), the task priority (PRIORITY)

The PILOT object properties that you can use to get values from Datacap and provide them to actions are.

- BATCHID
- BATCHDIR
- OPERATOR
- STATION
- CHILDRENQUANTITY
- PRIORITY
- PROJECTPATH
- CAPTION
- PAGESINBATCH
- DOCSINBATCH
- EXPECTEDPAGES
- ADJUSTEDPAGES
- ADJUSTEDDOCS
- JOBNAME
- FORMPROFILE
- DCOFILE
- JOBID
- TASKID

Example

In this example, the smart parameter returns the input DCO file for the current task.

Action and return value	XML example (if applicable)
-------------------------	-----------------------------

Action and return value	XML example (if applicable)
Action: rr_Get("PILOT(DCOFILE)")	
Return value: C:\Datacap\TravelDocs\batches\20110004.004\Export.xml	

Parent topic: [Miscellaneous special variables](#)

@PROJECTDIR

Description

Returns the path and file name for the setup (.xml) file of the current task. The path is relative to the *dco_folder of the application*.

Example

Action and return value	XML example (if applicable)
Action: rr_Get("@PROJECTDIR")	
Return value: \RRS_VScan.bpp	

Parent topic: [Miscellaneous special variables](#)

@PROCESSDIR

Description

Returns the full path to the *dco_<app_name>* folder of the application.

Example

Action and return value	XML example (if applicable)
Action: rr_Get("@PROCESSDIR")	
Return value: C:\Datacap\TravelDocs\dco_TravelDocs	

Parent topic: [Miscellaneous special variables](#)

@STRING(<string_value>)

Description

Returns the value that is specified as a string.

Example

Action and return value	XML example (if applicable)
Action: rr_Get("@STRING(MyString)")	
Return value: MyString	

@TIME(<format>)

Description

Returns the current time in the format specified. If you do not specify a format like @TIME(), defaults to HH:MM:SS.

Example

Action and return value	XML example (if applicable)
Action: rr_Get("@TIME(HH:MM)")	
Return value: 10:45	

Parent topic: [Miscellaneous special variables](#)

@TYPE

Description

Returns the type of the current object (Batch, Document, Page, or Field)

Example

Action and return value	XML example (if applicable)
Action: rr_Get("@TYPE")	
Return value: Page	

Parent topic: [Miscellaneous special variables](#)

Smart Parameter Build control

In Datacap Studio, you can use the Smart Parameter Build control to create an expression that is evaluated at run time that allows you to obtain data from Datacap objects and variables. In Datacap Studio, when you select an action that takes a string parameter, the Parameters section displays the details of the selected action along with a button that displays three dots. By using this button, you can open Smart Parameter Build control.

To create smart parameter expression, complete the following steps:

1. In the Smart Parameter Build window, from the Category list, select a smart parameter category. For example, Batch, Configuration, Navigation, and Objects.

When you select a category, the Element field displays all the elements that belong to the selected category.

2. Select an element.

When you select an element, the Preview field displays a smart parameter. For example, if you select Batch Identifier element, the Preview field displays the @BATCHID smart parameter.

3. To add the smart parameter to the expression, click the Add button.

The Smart Parameter Expression field displays the parameter. You can change the smart parameter expression in the Smart Parameter Expression field, and add text that you want.

4. Under the Evaluation section, click the Evaluate button.

Note: When you click the Evaluate button, Smart Parameter Build attempts to evaluate the current smart parameter expression. However, the results you get can be different at run time. For example, when Smart Parameter Build tries to evaluate @BATCHID, there is no batch running so there is no Batch ID for it. So interactively, Smart Parameter Build returns a sample Batch ID after you click the Evaluate button.

Other components are not evaluated so cannot return values. For example, if the expression is @B.MyRuntimeVariable, it does not exist at setup time so Smart Parameter Build evaluates to an empty string at setup time. At runtime, Smart Parameter Build evaluates whatever the variable contains.

5. Click Save Expression, the Parameters section in Desktop Studio displays the smart parameter. For example, @BATCHID.

You can also specify two or multiple smart parameters in an expression. To specify two or more parameters, select another element from the Element list, for example, Job Name, and then click Add. The Smart Parameter Expression field displays the value as @BATCHID+@JOBNAME. You can specify values such as @BATCHID:+:@JOBNAME.

Many objects are created at run time so the Smart Parameter Builder cannot show the objects because they do not exist, and to access these runtime only variables and objects, you must manually type the objects that you want to access at run time.

The Smart Parameter Build control runs at setup time so the DCO objects might not exist and the evaluation result can be different at run time. You must edit expressions manually to access objects only available at run time or to add your own custom text. You can use a "+" sign to separate unique smart parameters.

For the Navigation category, the Elements field displays the elements that belong to Navigation category. In this case, the setup time and runtime become apparent. For example, if you select the Page element, the Object field displays subobjects. The list displays the subobjects that are part of the DCO tree in Datacap Studio at setup time.

Select a subobject, such as Object Shipping, and click Navigate. The Smart Parameter Builder window displays the corresponding field, such as Navigated to: Object Shipping. You can select any one of the displayed variables of Page object, and click Add.

At run time, on a typical application various variables are added by using actions or application. However, the variables do not exist at setup time.

Parent topic: [Smart Parameter Special Variable Reference](#)

Standard Variable Reference

Standard variables can be used on batches, documents, pages, and fields.

- [Variables that are used on all object types](#)
Some variables are available for all object types.
- [Batch variables](#)
These standard variables can be used only on batch object types.
- [Document variables](#)
These standard variables can be used only on document object types.

- [Page variables](#)
These standard variables can be used only on page object types.
- [Field variables](#)
These standard variables can be used only on field object types.

Variables that are used on all object types

Some variables are available for all object types.

- [MAX_TYPES](#)
- [MESSAGE](#)
- [MIN_TYPES](#)
- [rules](#)
- [STATUS](#)
- [TYPE](#)
- [hr_locale](#)
The hr_locale variable specifies the locale value that is used by the recognition engines to validate currency, numerals, and date data types for localization.
- [The hr_SyncImg variable](#)
The hr_SyncImg variable affects the Click N Key during the verification task.

Parent topic: [Standard Variable Reference](#)

MAX_TYPES

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Description

Specifies the maximum number of child object types that can be present at run time to meet document integrity requirements (0 = no maximum).

Parent topic: [Variables that are used on all object types](#)

MESSAGE

Applies to

Applies	Does not apply
Runtime DCO	Setup DCO

Description

Used by many validation and look up actions to report errors.

Example

This example shows an error message that is written to the runtime hierarchy by a failed validation action.

```
<V n="MESSAGE">Failed Calculation:377.73=477.73</V>
```

Parent topic: [Variables that are used on all object types](#)

MIN_TYPES

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Description

Specifies the minimum number of child object types that must be present at runtime to meet document integrity requirements.

Parent topic: [Variables that are used on all object types](#)

rules

Applies to

Applies to: Setup DCO Does not apply: Runtime DCO

Description

Stores the rule map of an object that is established in Datacap Studio.

For example, you can map one or more rules to an object in the Document hierarchy. When you select the rule in the Document hierarchy object to which you mapped the rule, Datacap Studio displays the DCO.Rule Map in the Properties tab. The values in the DCO.Rule Map include the Ruleset ID and the Rule name. The Ruleset ID references the position of the ruleset on the Rulesets tab in Datacap Studio, and the Rule name references the position of the rule in the ruleset. For example, if an object is mapped with a rule named Vendor, which is part of the Validate ruleset, then the Rule name value can be 4 and the Ruleset ID can be 6. These values mean that the Validate ruleset is in the sixth position in the Ruleset tab, and the Vendor rule is in the fourth position within the Validate ruleset. The corresponding line in the Setup DCO is `<r`

`id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar070_4"`
`rs="6" />`. If the object is mapped with five different rules, then the Setup DCO can be similar to this sample:

```
<in>
<r
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar070_4"
rs="6" />
<r
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar070_2"
rs="13" />
<r
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar070_9"
rs="7" />
<r
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar070_3"
rs="14" />
<r
```

```
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcvar070_6"
rs="16" />
</in>
```

Example

This example shows how the XML in the sample is stored in the Setup DCO file.

```
<V n="rules"><in><r id="4">
rs="6" /><r id="2" rs="13"
/><r id="9" rs="7" /><r
id="3" rs="14" /><r
id="6" rs="16" /></in></V>
```

Parent topic: [Variables that are used on all object types](#)

STATUS

Applies to

Setup DCO, Runtime DCO

Description

Specifies the status of the object. The status is initially zero or not present in the Setup DCO. The status is updated at run time depending on the object type and the rules and actions of the task. Some common status values are shown in this table with their conventional meanings. Specific applications can use other status values.

Value	Status	Applies to	Assigned by
49	ScanOK	Page objects	Scan tasks
0	OK	Batch, document, and page objects	Rulerunner tasks
1	Problem		
2	Overridden	Pages that fail validation but are overridden by the operator.	Verify tasks
48	RecogDoneOK	Page objects	Recognition tasks
0	OK	Field objects	Any task
1	Error		Can specify -1 in Setup DCO
-1	Ignore		

A few less common status values are shown in this table.

Value	Status
51	CannotFindAnchor
52	DontNeedVerification
70	RescanPage

Value	Status
71	VerificationFailed
72	PageOnHold
73	PageOverridden
74	NoData
75	DeletedPage
77	DeleteApproved
79	ReviewPage
128	DeletedDoc
145	ReviewDoc

Example

```
<V n="STATUS">1</V>
```

Parent topic: [Variables that are used on all object types](#)

TYPE

Applies to

Applies	Does not apply
Setup DCO	
Runtime DCO	

Description

In the Setup DCO, TYPE specifies the object type (batch, document, page, or field), for example:

```
<V n="TYPE">Page</V>
```

TYPE is updated in the Runtime DCO to specify the object name, for example:

```
<V n="TYPE">Rental_Agreement</V>
```

Parent topic: [Variables that are used on all object types](#)

hr_locale

The hr_locale variable specifies the locale value that is used by the recognition engines to validate currency, numerals, and date data types for localization.

Applies to

Global DCO

Description

The `hr_locale` variable uses the BCP 47 defined language values for language-region, language code ID, and locale description. For example, the `hr_locale` variable for the English language in the United States, reads `en-US, 1033, English (United States)`.

The value of this variable is set in terms of the DCO hierarchy. For example, if this variable is set at the batch level, it is also automatically set at the document and page level. For more information about the hierarchy, see [Relation of the document hierarchy to the runtime batch hierarchy](#).

Parent topic: [Variables that are used on all object types](#)

The `hr_SyncImg` variable

The `hr_SyncImg` variable affects the Click N Key during the verification task.

Applies to

The `hr_SyncImg` applicable only when dealing with multi page documents where the data might be located on a trailing page.

Description

By setting this variable to the appropriate value the operator can locate a field on the trailing page. If this variable is not set, or is equal to zero, you cannot draw any fields on a trailing page, because as soon as you click a field in the verify panel while a trailing page is in image view, the displayed image changes and the first page is displayed in an image view.

You can use the following possible values for this variable:

- 0 - Do not sync
- 1 - Sync always
- 2 - Sync only when field position is not empty or all zeros

Parent topic: [Variables that are used on all object types](#)

Batch variables

These standard variables can be used only on batch object types.

- [LAST_RR_PROFILE](#)
- [ScanSrcChannel](#)

Parent topic: [Standard Variable Reference](#)

LAST_RR_PROFILE

Applies to

Does not apply: Setup DCO Applies to: Runtime DCO

Description

Specifies the name of the last task profile that ran.

Example

```
<V n="LAST_RR_TPROFILE">Rulerunner:m:eRun</V>
```

Parent topic: [Batch variables](#)

ScanSrcChannel

Applies to

Does not apply: Setup DCO Applies to: Runtime DCO

Description

The input channel of the batch. For example: the input channel can be a fax or scanner, or some other application-specific source. If your application sets this variable before the Export Statistics ruleset is run, the Datacap Dashboard will use the value to group the batch in the Dashboard. If your application does not set *ScanSrcChannel*, the Export Statistics ruleset uses the current Job Name as the value for the source.

Example

```
<V n="ScanSrcChannel">Fax 213-555-7001</V>
```

Parent topic: [Batch variables](#)

Document variables

These standard variables can be used only on document object types.

- [DD](#)

Parent topic: [Standard Variable Reference](#)

DD

Applies to

Runtime DCO

Description

Used by some scan tasks - DD contains the value that is imprinted on the first page of the document by a scanner, or an externally assigned document ID.

Parent topic: [Document variables](#)

Page variables

These standard variables can be used only on page object types.

- [Confidence](#)

- [DATAFILE](#)
- [Fingerprint Created](#)
- [Image_Offset](#)
- [IMAGEFILE](#)
- [Latitude](#)
- [Longitude](#)
- [PAGE_HEIGHT](#)
- [PAGE_WIDTH](#)
- [PageName](#)
- [PatternConfidence](#)
- [PD](#)
- [ScanSrcInputFolder](#)
- [ScanSrcFileName](#)
- [ScanSrcSubFolder](#)
- [ScanSrcPath](#)
- [TEMPLATE IMAGE](#)
- [TemplateID](#)

Parent topic: [Standard Variable Reference](#)

Confidence

Applies to

• Setup DCO | [Runtime DCO](#)

Description

Specifies the confidence level achieved during fingerprint matching.

Parent topic: [Page variables](#)

DATAFILE

Applies to

[Runtime DCO](#)

Description

Specifies the name of the data XML file that is associated with the page. DATAFILE is initially blank in the Setup DCO and is assigned at run time (for example, TM000001.xml).

Parent topic: [Page variables](#)

Fingerprint Created

Applies to

[Runtime DCO](#)

Description

Specifies whether or not Datacap added a fingerprint for the page to the fingerprint library, which happens when fingerprint matching fails and the argument to the FindFingerprint action is True.

Example

```
<V n="Fingerprint Created">No</V>
```

Parent topic: [Page variables](#)

Image_Offset

Applies to

Runtime DCO

Description

Specifies the offset in pixels (x, y) between the runtime page image and the fingerprint image.

Example

```
<V n="Image_Offset">-100,-100</V>
```

Parent topic: [Page variables](#)

IMAGEFILE

Applies to

- Setup DCO
- Runtime DCO

Description

Specifies the name of the associated runtime image file. IMAGEFILE is blank in the Setup DCO and is assigned at run time.

Example

```
<V n="IMAGEFILE">tm000010.tif</V>
```

Parent topic: [Page variables](#)

Latitude

Applies to

Applies	Does not apply

Description

When capturing images on a mobile device, the Latitude variable stores the latitude value of the GPS location at capture time.

Parent topic: [Page variables](#)

Longitude

Applies to

Applies	Does not apply
Runtime DCO	Setup DCO

Description

When capturing images on a mobile device, the Longitude variable stores the longitude value of the GPS location at capture time.

Parent topic: [Page variables](#)

PAGE_HEIGHT

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Description

This variable is used for automatic zonal OCR to store the height of the image for the last page type on which a zone was drawn.

Parent topic: [Page variables](#)

PAGE_WIDTH

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Description

This variable is used for automatic zonal OCR to store the width of the image for the last page type on which a zone was drawn.

Parent topic: [Page variables](#)

PageName

Applies to

Applies	Does not apply
Runtime DCO	Runtime DCO

Description

The PageName variable is used by scan tasks only and specifies the file name without the path or the file extension. The variable parses the ScanSrcPath variable, and selects a substring between the last slash and the first period after that.

Here is an example with multiple periods in the file name, and PageName ignores all characters after the first period:

```
<V n="ScanSrcPath">C:\datacap\APT\Images\Input\APT003.filename3.tif</V>
```

```
<V n="PageName">APT003</V>
```

Parent topic: [Page variables](#)

PatternConfidence

Applies to

Applies	Does not apply
Runtime DCO	Setup DCO

Description

Specifies the confidence level achieved when pattern matching is used for page identification.

Example

```
<V n="PatternConfidence">10</V>
```

Parent topic: [Page variables](#)

PD

Applies to

Applies	Does not apply

Runtime DCO	Setup DCO
-------------	-----------

Description

Used by isscan tasks only - specifies the page data string.

Parent topic: [Page variables](#)

ScanSrcInputFolder

Applies to

Applies	Does Not Apply
Runtime DCO	Setup DCO

Description

The full path of the input folder for this file. This variable is set indirectly by the mv_retain_folder action.

Example

```
<V n="ScanSrcInputFolder">c:\shared\group1</V>
```

Parent topic: [Page variables](#)

Related reference:
[mv_retain_folder](#)

ScanSrcFileName

Applies to

Applies	Does Not Apply
Runtime DCO	Setup DCO

Description

The original name of the file without the path. This variable is set indirectly by the mv_retain_folder action.

Example

```
<V n="ScanSrcFileName">invoice_0001.tif</V>
```

Parent topic: [Page variables](#)

Related reference:
[mv_retain_folder](#)

ScanSrcSubFolder

Applies to

Applies	Does Not Apply
Runtime DCO	Setup DCO

Description

The relative path of the input folder in which the file was found, which includes the root input folder name but not the root folder path. This variable is set indirectly by the mv_retain_folder action.

Example

```
<V n="ScanSrcSubFolder">\shared\group1\batch1</V>
```

Parent topic: [Page variables](#)

Related reference:

[mv_retain_folder](#)

ScanSrcPath

Applies to

Applies	Does Not Apply
Runtime DCO	Setup DCO

Description

Used by scan tasks only - specifies the full path and file name to the original image file.

Example

```
<V n="ScanSrcPath">c:\datacap\apt\images\input\invoice_0001.tif</V>
```

Parent topic: [Page variables](#)

TEMPLATE IMAGE

Applies to

Setup DCO and Runtime DCO

Description

Specifies the name of the matching fingerprint (CCO) file. The value is blank in the Setup DCO and is assigned at runtime.

Parent topic: [Page variables](#)

TemplateID

Applies to

Applies	Does not apply
Runtime DCO	Setup DCO

Description

Specifies the ID of the matching fingerprint.

Example

```
<V n="TemplateID">567</V>
```

Parent topic: [Page variables](#)

Field variables

These standard variables can be used only on field object types.

- [DataType](#)
- [DensityString](#)
- [DICT](#)
- [Index](#)
- [label](#)
- [Location](#)
- [Lookup](#)
- [LookupEx](#)
- [MaxLength](#)
- [METRIC](#)
- [MultiLine](#)
- [MultiPunch](#)
- [PatternMatch](#)
- [PictureString](#)
- [Pos<templateID>](#)
- [Position](#)
- [ReadOnly](#)
- [RecogStatus](#)
- [RecogType](#)
- [ReqConf](#)
- [SELECT](#)
- [ShowChar](#)
- [Sticky](#)
- [Text](#)
- [Zone_Offset](#)

Parent topic: [Standard Variable Reference](#)

DataType

Applies to

Setup DCO

Verification panels that support this variable

Datacap Navigator

Datacap Web Client (VeriFine.aspx)

Description

Specifies the type of characters the user can enter into the field in the Verify panel. Use the following codes to specify the allowed data type:

Value	Description
0	Alphanumeric
1	Integer
2	Float
3	Date
4	Time
5	Currency

Parent topic: [Field variables](#)

DensityString

Applies to

Runtime DCO

Description

This variable is used for OMR zones, one character per zone, where the character represents the percentage of black pixels within the zone according to the following formula.

Character's ASCII code value = Percentage black pixels + 48

For example, if the zone has 20% black pixels, the result is ASCII code 68 = 'D'.

Example

This example represents a field with three OMR check boxes.

```
<V n="DensityString">@@D</V>
```

Parent topic: [Field variables](#)

DICT

Applies to

Setup DCO

Verification panels that support this variable

Datacap Desktop

Datacap Navigator

Datacap Web Client (VeriFine.aspx and aindex.aspx)

Description

Used with selection fields and specifies the name of a dictionary (within this Setup DCO) containing a list of possible values.

Parent topic: [Field variables](#)

Index

Applies to

Runtime DCO

Description

Used in FormSpec to optionally specify the field's index.

Parent topic: [Field variables](#)

label

Applies to

Setup DCO

Verification panels that support this variable

- Datacap Desktop
- Datacap Navigator
- Datacap Web Client (VeriFine.aspx)
- Datacap Web Client (aindex.aspx)

Description

The value of this variable, if specified, defines the label that is displayed beside the field in the supported verification panels. If not specified, the `type` attribute of the field is used as the label.

Parent topic: [Field variables](#)

Location

Applies to



Applies	Does not apply
Runtime DCO	Setup DCO

Description

When capturing images on a mobile device, the Location variable stores the address of the GPS location at capture time.

Parent topic: [Field variables](#)

Lookup

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Verification panels that support this variable

Support this variable	Do not support this variable
Datacap Desktop	N/A
Datacap Navigator	
Datacap Web Client (VeriFine.aspx)	
Datacap Web Client (aindex.aspx)	

Description

Specifies a database lookup statement that gets run during verification when the user clicks the hyperlinked field label. A list of matching entries from the database that is specified by the dns attribute is displayed. The selected entry is used to populate the fields that are specified in the flist attribute.

Example

The following sample Lookup value gets a list of car types from the lookup database that is specified in the application configuration (.app) file. The sample code then populates the Car_Type field with the selected value.

```
<SQL flist='Car_Type' dsn="*/lookupdb:cs">SELECT Car_Type FROM Car_Types</SQL>
```

The next example (from APT) gets a list of matching vendor names, ZIP codes, and vendor IDs from the application's lookup database. The list is then displayed to the user. The SQL statement uses the text in the Vendor field as the search string. The user can then, for example, enter the first letter and see a list of vendor names that start with that letter. Upon selecting a vendor from the list, it populates the Vendor, Remittance_Zip, and Vendor_Number fields with the information for the selected vendor.

Attention: The carriage return in the following example is only for readability; it is not technically required.

```
<SQL flist='Vendor,Remittance_Zip,Vendor_Number' dsn="*/lookupdb:cs">
SELECT VendorName,VendorZip,VendorID FROM VendorTable WHERE VendorName LIKE
```

```
'@@Vendor@@%'</SQL>
```

Note in this example the special syntax that is required to reference a field value from the SQL statement: @@Vendor@@%. Note also that WHERE <column> = '<value>' is not supported.

Parent topic: [Field variables](#)

LookupEx

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Verification panels that support this variable

Support this variable:	Do not support this variable:
Datacap Web Client (VeriFine.aspx)	Datacap Desktop
	Datacap Navigator
	Datacap Web Client (aindex.aspx)

Description

Specifies a database lookup statement that gets run during verification when the user leaves the field (for example, by clicking or moving to the next field). LookupEx is typically used to populate other fields that are based on current field's value. The structure of the lookup statement is similar to that of the Lookup variable.

Example

The sample LookupEx value looks up the vendor name that is based on the ID in the VendorID field and populates the VendorName field with the result.

Attention: The carriage return in the following example is only for readability; it is not technically required.

```
<SQL flist='VendorName' dsn="*/lookupdb:cs">  
SELECT Vendor FROM VendorTable WHERE VendorID LIKE '@@VendorID@%'</SQL>
```

For more information about the LookupEx variable, see the [Lookup](#) variable.

Parent topic: [Field variables](#)

MaxLength

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Verification panels that support this variable

Support this variable	Do not support this variable
Datacap Web Client (VeriFine.aspx)	Datacap Desktop Datacap Navigator Datacap Web Client (aindex.aspx)

Description

Specifies the maximum number of characters the user can type into the field in the Verify panel.

Parent topic: [Field variables](#)

METRIC

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Description

Specifies the size of the search region that is used during geometric pattern matching. The dimensions that are specified are relative to the anchor field. For example, METRIC=200,300 creates a search region 200 pixels larger left and right and 300 pixels larger above and below.

Parent topic: [Field variables](#)

MultiLine

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Verification panels that support this variable

Support this variable	Do not support this variable
Datacap Desktop	Datacap Navigator Datacap Web Client (VeriFine.aspx) Datacap Web Client (aindex.aspx)

Description

When set to '1,' Multiline shows the field as a multiline edit field in the Datacap Desktop Verify pane.

Parent topic: [Field variables](#)

MultiPunch

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Verification panels that support this variable

Support this variable	Do not support this variable
Datacap Desktop	Datacap Web Client (aindex.aspx)
Datacap Navigator	
Datacap Web Client (VeriFine.aspx)	

Description

Used when the field contains multiple OMR (check box) options. When MultiPunch is set to '1,' multiple selections are allowed; otherwise only one selection is allowed.

Parent topic: [Field variables](#)

PatternMatch

Applies to

Applies to	Does not apply
Setup DCO	Runtime DCO

Description

Some PatternMatch actions require anchor fields, which are used to adjust the image registration. You can identify one or more fields as an anchor field in the Zones on Datacap Studio.

To identify a field as an anchor field, lock the DCO, right-click the anchor field, and select Manage Variables. Create the variable PatternMatch, if it does not exist, and set the value to 1.

When the PatternMatch variable is set to 1, the field is an anchor field for pattern matching.

Parent topic: [Field variables](#)

PictureString

Applies to

Applies	Does not apply
---------	----------------

Verification panels that support this variable

Support this variable	Do not support this variable
Datacap Web Client (VeriFine.aspx)	Datacap Desktop Datacap Navigator Datacap Web Client (aindex.aspx)

Description

Specifies which characters the user can type into the field according to this key.

Value	Characters allowed
A	Alphabetic or space
a	Alphabetic, punctuation, or space
D	(Date) numeric digit, minus sign, decimal point (period), forward slash
F	(Float) numeric digit, minus sign, or decimal point (period)
f	Numeric digit or punctuation
L	Lowercase alphabetic or space
l	Lowercase alphabetic, punctuation, or space
N	Numeric digit
n	Uppercase alphabetic character, numeric digit, or space.
P	Punctuation or space
T	(time) numeric digit, A, P, M, or colon
U	Uppercase alphabetic or space
u	Uppercase alphabetic, punctuation, or space
X	Alphabetic, numeric digit, or space
x	Alphabetic, numeric digit, punctuation, or space
Z	Any character
#	Numeric digit or minus sign

Example

- `PictureString="A"` - Any upper/lower case letter or space is allowed (no numbers or special characters)
- `PictureString=""` - All characters are allowed (default)

Tip: If you specify more than one-character set code, the first code applies to the first character that is typed by the user. The second code applies to the second character that is typed by the user, a third code applies to the third character that is typed, and so on. The last code applies to all remaining characters that are typed.

Parent topic: [Field variables](#)

Pos<templateID>

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Description

Specifies the position of the recognition zone for a specific fingerprint image (< templateID>). This variable uses the upper left and lower right corners of the zone (x1, y1, x2, y2) to specify the position.

Parent topic: [Field variables](#)

Position

Applies to

Applies
Setup DCO
Runtime DCO

Verification panels that support this variable

Datacap Navigator

Description

Specifies the field's position by using the upper left and lower right corners (x1, y1, x2, y2). The value of Position is initially (0,0,0,0) in the Setup DCO and is populated at run time.

Parent topic: [Field variables](#)

ReadOnly

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Verification panels that support this variable

Support this variable	Do not support this variable
Datacap Web Client (VeriFine.aspx)	Datacap Desktop

	Datacap Navigator Datacap Web Client (aindex.aspx)
--	---

Description

When set to "1," the field is read-only in the Verify panel and the user cannot change the preset or recognized value.

Parent topic: [Field variables](#)

RecogStatus

Applies to

Does not apply: Setup DCO Applies to: Runtime DCO

Description

Numeric code set by some recognition actions to indicate status of the operation.

Parent topic: [Field variables](#)

RecogType

Applies to

Applies to: Setup DCO Does not apply: Runtime DCO

Verification panels that support this variable

Datacap Navigator

Description

Specifies the code for the recognition engine to use when reading data from this field. OMR (check box) fields require RecogType=4 and these fields are the only ones that typically require this variable.

Parent topic: [Field variables](#)

ReqConf

Applies to

Applies to: Setup DCO Does not apply: Runtime DCO

Verification panels that support this variable

Datacap Navigator

Description

Specifies the field level recognition confidence that is required for all characters that are populated to the field from a recognition action. Used with validation actions to flag the field for validation if any character in the recognized result is lower than this value.

Parent topic: [Field variables](#)

SELECT

Applies to

Applies	Does Not Apply
Setup DCO	Runtime DCO

Verification panels that support this variable

Support this variable:	Do not support this variable:
Datacap Desktop	Datacap Web Client (aindex.aspx)
Datacap Navigator	
Datacap Web Client (VeriFine.aspx)	

Description

Specifies a database lookup statement that converts an edit field in a verification panel into a drop-down list with values from a database. A list of matching entries from the database that is specified by the dns attribute is displayed. The selected entry is used to populate the fields that are specified in the flist attribute.

Example

This sample Lookup value retrieves a list of car types from the lookup database that is specified in the application configuration (.app) file. This sample then populates the list in the Car_Type field.

```
<SQL flist='Car_Type' dsn="*/lookupdb:cs">SELECT Car_Type FROM Car_Types</SQL>
```

Note that you can populate multiple fields simultaneously (see the Lookup variable for an example).

If you also configure the DICT variable, DICT variable values are displayed in the drop-down list at runtime in Datacap Navigator.

Parent topic: [Field variables](#)

ShowChar

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Verification panels that support this variable

Support this variable	Do not support this variable
Datacap Web Client (VeriFine.aspx)	Datacap Desktop Datacap Navigator Datacap Web Client (aindex.aspx)

Description

This variable works with the Datacap Web Client (VeriFine.aspx) panel only. When set to 1, the character zones are displayed in the image snippet of the field in the Verify panel.

Parent topic: [Field variables](#)

Sticky

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Verification panels that support this variable

Support this variable:	Do not support this variable:
Datacap Navigator	Datacap Desktop
Datacap Web Client (VeriFine.aspx)	Datacap Web Client (aindex.aspx)

Description

If Sticky is defined for any field (the value is not important), the Verify panel includes a check box to the left of the field's label. If you select the check box, enter a value, and submit the page, the field value populates across all pages of that type in the current batch.

Note: The Sticky variable does not support drop-down menu lists.

Parent topic: [Field variables](#)

Text

Applies to

Applies	Does not apply
Setup DCO	
Runtime DCO	

Verification panels that support this variable

Support this variable:	Do not support this variable:
Datacap Navigator	Datacap Desktop
Datacap Web Client (aindex.aspx)	Datacap Web Client (VeriFine.aspx)

Description

If defined for a field (the value must be empty), the field in the Verify panel becomes sticky. If the user enters a value and submits a page, the field value populates across all pages of that type in the current batch. The field must be initially empty for automatic population to work.

Parent topic: [Field variables](#)

Zone_Offset

Applies to

Applies	Does not apply
Setup DCO	Runtime DCO

Description

Used with anchor fields during geometric pattern matching. Specifies the offset in pixels (x, y) between the anchor's position on the runtime page image and its position in the fingerprint image.

Parent topic: [Field variables](#)

Application-specific variable reference

Some variables have been designed for use only with specific Datacap applications.

- [Medical Claims 5010 form configuration parameters](#)
The configuration variables for the Medical Claims 5010 Institutional and Professional forms can be found in the configuration .ini files (5010_Inst.ini and 5010_Prof.ini).

Medical Claims 5010 form configuration parameters

The configuration variables for the Medical Claims 5010 Institutional and Professional forms can be found in the configuration .ini files (5010_Inst.ini and 5010_Prof.ini).

- [5010 Institutional form configuration variables](#)
The following variables are configured in the 5010_Inst.ini configuration file.
- [5010 Professional form configuration variables](#)
The following variables are configured in the 5010_Prof.ini configuration file.

Parent topic: [Application-specific variable reference](#)

5010 Institutional form configuration variables

The following variables are configured in the 5010_Inst.ini configuration file.

Table 1. General

Variable	Default	Description
Implied Decimal	TRUE	When set to TRUE, this boolean variable normalizes all currency data values to have a two-digit value for cents such as 0.00. The variable then formats output for 837 requirements of no leading or trailing zeros.
PreFilter	TRUE	This boolean variable filters character output to EDI elements when set to TRUE. The only characters allowed are Comma, Hyphen, Period, 0-9, and A-Z. Lower case a-z characters are capitalized.

Table 2. Log

Variable	Default	Description
SegmentLog	TRUE	When set to TRUE, this boolean variable adds a single log entry to the RRS log when a segment is counted for adding to the EDI export stream.
SegmentReport	FALSE	When set to TRUE, this variable saves all segment logs added to the export stream and outputs the stream to the RRS log file as a summary after the EDI is created. This variable does not affect output.

Table 3. X12N

Variable	Default	Description
SegmentSeparator	*	Segment Separator character which indicates the start/end of 837 segment elements.
SegmentTerminator	~	Segment Terminator character which indicates the end of a single 837 segment.
RepetitionSeparator	^	Delimiter character used to separate repeated occurrences of a simple data element or composite data structure.

Vari abl e	D e f a u l t	Description
Ter min ator Plu sLin eFe ed	F A L S E	When set to TRUE , this variable adds a line feed and carriage return to the end of each Segment Terminator character. Typically this option is disabled in a production environment, and is only used for debugging the output data. Attention: The program will by default also create a file named EDI_OutputText.txt which performs the same debugging function. This file is generated using the output file to ensure data integrity, and cannot be disabled.
Sup pre ssS Vda teS pan	F A L S E	When this variable is set to TRUE , there is no creation of a DTP RD8 date span in the service line even if Type of Bill is Inpatient type.
AllU ppe rCa se	F A L S E	When set to TRUE , all output characters are converted to uppercase. Restriction: Use of the Prefilter option forces all characters to uppercase regardless of this option.
App end ToL ast GS GE	F A L S E	When merging individual claim EDI files, this option chooses whether the new claim is added as the first claim in a new GS/GE loop in the merged file. When this option is set to FALSE , each claim will be created in a new GS/GE and ST/SE loop in the merged file. When this option is set to TRUE , then each claim will be added to the existing GS/GE loop in the merged file. See the AppendToLastSTSE option for ST/SE information when added with this option set to TRUE .
App end ToL ast STS E	F A L S E	When merging individual claim EDI files, this option chooses whether the new claim is added as the first claim in a new ST/SE loop in the merged file. When this option is set to FALSE , each claim will be created in a new ST/SE loop within the existing GS/GE loop in the merged file. When this option is set to TRUE , then each claim will be added to the existing GS/GE loop in a new ST/SE loop in the merged file.

Table 4. ISA_Header

Vari able	D e f a u l t	Description
Auth oriza tionI nfor mation		Sets value for ISA02. Limited to ten characters, empty by default. A value placed here will automatically set ISA01 from 00 (default-No authorization info present) to 03 (Additional Data Identification).

Variable	Default	Description
SecurityInformation		Sets value for ISA04. Limited to ten characters, empty by default. A value placed here will automatically set ISA03 from 00 (default-No Security Info present) to 01 (Password).
InterchangeSenderQualifier	ZZ	Sets value for ISA05. Code indicating the type of Sender ID. Value codes are: 01, 14, 20, 27, 28, 29, 30, 33 or ZZ. Standard setting is ZZ.
InterchangeSenderID	DACAAP	Sets value for ISA06. Sender ID value, limited to 15 characters.
InterchangeReceiverQualifier	ZZ	Sets value for ISA07. Code indicating the type of Receiver ID. Value codes are: 01, 14, 20, 27, 28, 29, 30, 33 or ZZ. Standard setting is ZZ.
InterchangeReceiverID	00000	Sets value for ISA08. Receiver ID value, limited to 15 characters.
InterchangeControlNumber	BATCH	Sets the value for ISA13. Valid values are BATCH or a fixed value. Use of BATCH will use the current batch ID - right-justified and minus decimal characters. Any other entry is treated as a fixed output value for this element. This value is limited to fifteen numeric characters. Attention: By default, if the Batch ID is a split batch alphanumerical value, then the new value will be the 3 digit Julian Day, followed by the three digit daily batch number, and then concluded with the integer conversion of the alphanumerical split batch value. The maximum length of the split batch number is limited to 3 digits. Alternatively, the value for ISA(13) also supports smart parameters, so that entries can generate or create their own unique values and include it in the export.

Variable	Default	Description
AcknowledgmentRequested	0	Sets value for ISA14. Valid values are 0 (No Acknowledgement Default) or 1 (Interchange Acknowledgement Requested)
TestIndicator	P	Sets value for ISA15. Valid values are P (production) or T (test).
ComponentElementSeparator	<	Sets value for ISA16 the sub-element separator value. Segment Elements have multiple values are required to separate the values with a predefined character. Recommended are : or <.

Table 5. GS_Header

Variable	Default	Description
SenderCode	DATA CAP EXPORT	GS02 value limited to 15 characters.
ReceiverCode	00000	GS03 value limited to 15 characters, Defaults to 00000.
EDIStandard	005010X223A2	GS08 EDI version identifier code.

Table 6. Header

Variable	Default	Description
SubmitterName	The Defaults NM103 setting.	Sets the Loop 1000A NM1(03) value. Limited to 60 characters.
SubmitterPhone	The Defaults PER04 setting.	Sets the Loop 1000A PER(03) value.
SubmitterID	The GS_Header SenderCode setting value. If SenderCode is blank this value defaults to the Defaults NM109 setting.	Sets the Loop 1000A NM1(09) value. The standard setting is usually blank which fills this value with the GS_Header SenderCode value.
ReceiverName	The Defaults NM103 setting.	Sets the Loop 1000B NM1(03) value.
ReceiverID	00000	
SequenceSerialNumber	The value found in ST(02)	Sets the BHT(03) Reference ID value. Limited to 50 characters.

Variable	Default	Description
TransactionSetControlSource	The current Document ID value	Sets ST(02) value to use a Batch parameter. Entry is the parameter name to use as the source for the ST(02) value.
PayerIDNumber	123456789	Sets Loop 2000B SBR(03) value if field 11 Plan and Policy are empty. See also the CheckJobID settings in this setting section. Also sets Loop 2010BB NM1(09) if field 11 Policy number is empty. Note this value is overridden if the setting 2010BB_BC PrimaryPayerIDNumber has a value.
CheckJobID	FALSE	<p>When set to TRUE, this variable enables the ValueByJobID (Key) utility to retrieve values based on the JobID of the task. This allows the processing of claims for multiple entities, such as different payers.</p> <p>For example, the following function looks in the Header section for the number of different JobID's for which to check:</p> <pre>[Header] CheckJobID=TRUE/FALSE numJobID=2 numJobID[1]=JobName1 numJobID[2]=JobName2</pre> <p>Use values for paramter 'Key' (PayerIDNumber in example) as follows:</p> <pre>[JobName1] PayerIDNumber=1111111 [JobName2] PayerIDNumber=2222222</pre> <p>The example shows ini entries numJobID, numJobID[n] and JobName[n] PayerIDNumber that must be created by the user.</p>

Table 7. FileNameFormat

Variable	Default	Description
FileHeader	00501_	The default file name is the Page or Doc ID with decimals removed and spaces replaced with the underscore character. This is an option to place the specified text leading the default file name. For example if the Doc ID 2012004.03 is used and the FileHeader contains 00501_ then the file name becomes 00501_201200403. This can be used in conjunction with the FileTrailer option.
FileTrailer		The default file name is the Page or Doc ID with decimals removed and spaces replaced with the underscore character. This is an option to place the specified text trailing the default file name. For example if the Doc ID 2012004.03 is used and the FileTrailer contains _0500 then the file name becomes 201200403_0500. This can be used in conjunction with the FileHeader option.
MergeFileHeader	5010	If no File name is specified when calling to merge the 5010 file, the default file name becomes the batch Id. This is an option to place the specified text leading the default file name.
MergeFileTrailer		If no File name is specified when calling to merge the 5010 file, the default file name becomes the batch Id. This is an option to place the specified text trailing the default file name. The default value is blank.

Table 8. Defaults

Variable	Default	Description
NM103		General default for NM1(03) segment element values. Loops 1000A, 1000B, 2010BA, 2010CA, 2310A, 2310B, 2310C, 2330A, 2330B, 2420A.
NM104		General default for NM1(04) segment element values. Loops 2010BA, 2010CA, 2310A, 2310B, 2310CDF, 2330A, 2420A.
NM109		General default for NM1(09) segment element values. Loops 1000A, 1000B, 2010BA, 2310A, 2310CDF, 2330A, 2330B.
N301		General default for N3(01) segment element values. All instances.
N401		General default for N4(01) segment element values. All instances.
N402		General default for N4(02) segment element values. All instances.
N403		General default for N4(03) segment element values. All instances.

Variable	Default	Description
PER04		General default for PER(04) segment element values. Loop 1000A, 2010AA
DMG02		General default for DMG(02) segment element values. Loop 2000CA, 2010BA.
CLM07		Value for setting Loop 2300 CLM(07). Default is blank.
CLM08		Value for setting Loop 2300 CLM(08) if field 53 is not N or Y. Default is blank.
CLM09		Value for setting Loop 2300 CLM(09) if field 52 is not I or Y. Default is blank.
HI_BK		Default principle diagnosis code if no diagnosis fields on the claim are filled. Default is blank. Loop 2300 HI BK.
SV201		Default Loop 2400 SV2(1) RevCode if captured value is blank or all zeros. Default is blank.
OccDt		Default Date Value. Loop 2300 DTP 434 & 345, also Loop 2300 default date if procedure date is not valid, and occurrence span dates, Loop 2400 DTP 472

Table 9. ImpliedDecimalON

Variable	Default	Description
BuildLog	FALSE	TRUE/FALSE value. Adds RRS logging when building EDI if the ImpliedDecimal setting is enabled. See the General settings.

Table 10. 2000B

Variable	Default	Description
SubscriberIsPatient	FALSE	TRUE/FALSE. Setting for forcing EDI to output Subscriber is the Patient. Default is false.
UseCurrentPayer	See description	Value is a Regular expression that is tested against the value in field 50. If a match is found this is the Payer for this claim. Otherwise default is the last payer a, b or c on the completed form. Example: If payer is always Medicaid. Use of regular expression <code>MED[iI11T]iI11T]D</code> will find the correct field 50 line to assign. Default is blank and uses last payer on form.

Variable	Default	Description
UsePreviousPayer	Sequence	Value is a Regular expression that is tested against the value in field 50. If a match is found the line before the match becomes the Payer info for this claim. Otherwise default is the last payer a, b or c on the completed form. Example: If payer is always listed before Medicaid. Use of regular expression MED[<i>iI11T</i>]CA[<i>iI11T</i>]D will find the prior field 50 line to assign. Default is blank and uses last payer on form. Attention: Both UseCurrentPayer and UsePreviousPayer can be used simultaneously. Use Previous takes priority if both expressions match.

Table 11. 2010BB_BC

Variable	Default	Description
PrimaryPayerName		Setting to use for all processed claims for Loop 2010BB NM1(03). Takes priority over captured value as well as default value set in Header PayerName section.
PrimaryPayerIDNumber		Setting to use for all processed claims for Loop 2010BB NM1(09). Takes priority over default value set in Header PayerIDNumber section.
PrimaryPayerAddress		Value to place in Loop 2010BB N3(01).
PrimaryPayerCity		Value to place in Loop 2010BB N4(01).
PrimaryPayerState		Value to place in Loop 2010BB N4(02).
PrimaryPayerZip		Value to place in Loop 2010BB N4(03).

Table 12. 2300

Variable	Default	Description
PatientReason		Default value for Patient Reason for admission if field 69 is blank. Outpatient Claims only, default is blank.
DischargeStatus	01	Default value for Loop 2300 CL1(03) if field 17 is blank.
CalculateTotal	TRUE	TRUE/FALSE. Option to calculate the lineitem total using service line data instead of the field 28 total charges field for Loop 2300 CLM(02).

Table 13. 2310A

Variable	Default	Description
CreateOutpatient	FALSE	TRUE/FALSE. Setting to optionally always export Loop 2310A even if Type of Bill is outpatient.
ExportSecondaryID	FALSE	TRUE/FALSE. Optionally export a REF 1G segment in this loop with the value from field 76.

Table 14. 2320

Variable	Default	Description
OI03	Y	Value for Loop 2320 OI(03).
OI06	Y	Value for Loop 2320 OI(06).

Table 15. 2330A

Variable	Default	Description
UsePatientAddress	FALSE	TRUE/FALSE. Option to use always use field 9 address values for Loop 2330A N4 segment if patient is not the subscriber.
SubscriberCity		Default value for Loop 2330A N4(01) if patient is not the subscriber and UsePatientAddress is not enabled.
SubscriberState		Default value for Loop 2330A N4(02) if patient is not the subscriber and UsePatientAddress is not enabled
SubscriberZip		Default value for Loop 2330A N4(03) if patient is not the subscriber and UsePatientAddress is not enabled

Table 16. 2330B

Variable	Default	Description
PayerCity		Value to place in Loop 2330B N4(01).
PayerState		Value to place in Loop 2330B N4(02).
PayerZip		Value to place in Loop 2330B N4(03).
PayerIDNumber		Value to place in loop 2330B NM1(09) if field 51 is blank.

Table 17. N3

Variable	Default	Description
BuildLog	FALSE	TRUE/FALSE value. Adds more detailed RRS logging when creating N3 segments.
KeepTrailingSeparators	FALSE	TRUE/FALSE value. Option to not remove blank trailing elements from N3 segments.

Table 18. N4

Variable	Default	Description
BuildLog	FALSE	TRUE/FALSE value. Adds more detailed RRS logging when creating N4 segments.
KeepTrailingSeparators	FALSE	TRUE/FALSE value. Option to not remove blank trailing elements from N4 segments.

Table 19. REF

Variable	Default	Description
BuildLog	FALSE	TRUE/FALSE value. Adds more detailed RRS logging when creating REF segments.

Parent topic: [Medical Claims 5010 form configuration parameters](#)

5010 Professional form configuration variables

The following variables are configured in the 5010_Prof.ini configuration file.

Table 1. General

Variable	Default	Description
ImpliedDecimalON	TRUE	When set to TRUE, this boolean variable normalizes all currency data values to have a two-digit value for cents such as 0.00. The variable then formats output for 837 requirements of no leading or trailing zeros.
PreFilter	TRUE	This boolean variable filters character output to EDI elements when set to TRUE. The only characters allowed are Comma, Hyphen, Period, 0-9, and A-Z. Lower case a-z characters are capitalized.

Table 2. Log

Variable	Default	Description
SegmentLog	TRUE	When set to TRUE, this boolean variable adds a single log entry to the RRS log when a segment is counted for adding to the EDI export stream.
SegmentReport	FALSE	When set to TRUE, this variable saves all segment logs added to the export stream and outputs the stream to the RRS log file as a summary after the EDI is created. This variable does not affect output.

Table 3. X12N

Variable	Default	Description
SegmentSeparator	*	Segment Separator character which indicates the start/end of 837 segment elements.
SegmentTerminator	~	Segment Terminator character which indicates the end of a single 837 segment.

Variable	Default	Description
RepetitionSeparator	^	Delimiter character used to separate repeated occurrences of a simple data element or composite data structure.
TerminatorPlusLineFeed	FALSE	When set to <code>TRUE</code> , this variable adds a line feed and carriage return to the end of each Segment Terminator character. Typically this option is disabled in a production environment, and is only used for debugging the output data. Attention: The program will by default also create a file named EDI_OutputText.txt which performs the same debugging function. This file is generated using the output file to ensure data integrity, and cannot be disabled.
AllUpperCase	FALSE	When set to <code>TRUE</code> , all output characters are converted to uppercase. Restriction: Use of the Prefilter option forces all characters to uppercase regardless of this option.
AppendToLastGS/GE	FALSE	When merging individual claim EDI files, this option chooses whether the new claim is added as the first claim in a new GS/GE loop in the merged file. When this option is set to <code>FALSE</code> , each claim will be created in a new GS/GE and ST/SE loop in the merged file. When this option is set to <code>TRUE</code> , then each claim will be added to the existing GS/GE loop in the merged file. See the AppendToLastSTSE option for ST/SE information when added with this option set to <code>TRUE</code> .
AppendToLastSTSE	FALSE	When merging individual claim EDI files, this option chooses whether the new claim is added as the first claim in a new ST/SE loop in the merged file. When this option is set to <code>FALSE</code> , each claim will be created in a new ST/SE loop within the existing GS/GE loop in the merged file. When this option is set to <code>TRUE</code> , then each claim will be added to the existing GS/GE loop in a new ST/SE loop in the merged file.

Table 4. ISA Header

Variable	Default	Description
AuthorizationInformation		Sets the value for ISA02. Limited to ten characters, and blank by default. A value placed here will automatically set ISA01 from 00 (default-No authorization info present) to 03 (Additional Data Identification).
SecurityInformation		Sets the value for ISA04. Limited to ten characters, and blank by default. A value placed here will automatically set ISA03 from 00 (default-No Security Info present) to 01 (Password).

Variable	Default	Description
InterchangeSenderQualifier	Z Z	Sets the value for ISA05. This is a code indicating the type of Sender ID. Value codes are: 01, 14, 20, 27, 28, 29, 30, 33 or ZZ.
InterchangeSenderID		Sets value for ISA06. Sender ID value, limited to 15 characters.
InterchangeReceiverQualifier	Z Z	Sets the value for ISA07. This is a code indicating the type of Receiver ID. Value codes are: 01, 14, 20, 27, 28, 29, 30, 33 or ZZ.
InterchangeReceiverID		Sets value for ISA08. Receiver ID value, limited to 15 characters.
InterchangeControlNum		Sets the value for ISA13. Valid values are BATCH or a fixed value. Use of BATCH will use the current batch ID - right-justified and minus decimal characters. Any other entry is treated as a fixed output value for this element. This value is limited to fifteen numeric characters.
AcknowledgmentRequested		Sets the value for ISA14. Valid values are 0 (No Acknowledgement Default) or 1 (Interchange Acknowledgement Requested).
TestIndicator	P	Sets the value for ISA15. Valid values are P (production) or T (test).
ComponentElementSeparator	<	Sets the value for ISA16 the sub-element separator value. Segment elements that have multiple values are required to separate the values with a predefined character. Recommended values include : or <.

Table 5. GS_Header

Variable	Default	Description
SenderCode	DATA CAP EXPORT	GS02 value limited to 15 characters.
ReceiverCode	00000	GS03 value limited to 15 characters, Defaults to 00000.
EDIstandard	005010X222A1	GS08 EDI version identifier code.

Table 6. Header

Variable	Default	Description
SubmitterName	The Defaults NM103 setting.	Sets the Loop 1000A NM1(03) value. Limited to 60 characters.
SubmitterPhone	The Defaults PER04 setting.	Sets the Loop 1000A PER(03) value.

Variable	Default	Description
SubmitterID	The GS_Header SenderCode setting value. If SenderCode is blank this value defaults to the Defaults NM109 setting.	Sets the Loop 1000A NM1(09) value. The standard setting is usually blank which fills this value with the GS_Header SenderCode value.
ReceiverName	The Defaults NM103 setting.	Sets the Loop 1000B NM1(03) value.
ReceiverID	00000	
SequenceSerialNumber	The value found in ST(02)	Sets the BHT(03) Reference ID value. Limited to 50 characters.
TransactionSetControlSource	The current Document ID value	Sets ST(02) value to use a Batch parameter. Entry is the parameter name to use as the source for the ST(02) value.
PayerIDNumber	123456789	Sets Loop 2000B SBR(03) value if field 11 Plan and Policy are empty. See also the CheckJobID settings in this setting section. Also sets Loop 2010BB NM1(09) if field 11 Policy number is empty. Note this value is overridden if the setting 2010BB_BC PrimaryPayerIDNumber has a value.

Variable	Default	Description
CheckJobID	FALSE	<p>When set to TRUE, this variable enables the ValueByJobID (Key) utility to retrieve values based on the JobID of the task. This allows the processing of claims for multiple entities, such as different payers.</p> <p>For example, the following function looks in the Header section for the number of different JobID's for which to check:</p> <pre>[Header] CheckJobID=TRUE/FALSE numJobID=2 numJobID[1]=JobName1 numJobID[2]=JobName2</pre> <p>Use values for paramter 'Key' (PayerIDNumber in example) as follows:</p> <pre>[JobName1] PayerIDNumber=1111111 [JobName2] PayerIDNumber=2222222</pre> <p>The example shows ini entries numJobID, numJobID[n] and JobName[n] PayerIDNumber that must be created by the user.</p>

Table 7. FileNameFormat

Variable	Default	Description
FileHeader	0	The default file name is the Page or Doc ID with decimals removed and spaces replaced with the underscore character. This is an option to place the specified text leading the default file name. For example if the Doc ID 2012004.03 is used and the FileHeader contains 00501_ then the file name becomes 00501_201200403. This can be used in conjunction with the FileTrailer option.
FileTrailer	0	The default file name is the Page or Doc ID with decimals removed and spaces replaced with the underscore character. This is an option to place the specified text trailing the default file name. For example if the Doc ID 2012004.03 is used and the FileTrailer contains _0500 then the file name becomes 201200403_0500. This can be used in conjunction with the FileHeader option.

Variable	Default	Description
MergeFileHeader	5010-	If no File name is specified when calling to merge the 5010 file, the default file name becomes the batch Id. This is an option to place the specified text leading the default file name.
MergeFileTrailer		If no File name is specified when calling to merge the 5010 file, the default file name becomes the batch Id. This is an option to place the specified text trailing the default file name. The default value is blank.

Table 8. Defaults

Variable	Default	Description
NM103		General default for NM1(03) segment element values. Loops 1000A, 1000B, 2010BA, 2010CA, 2310A, 2310B, 2310C, 2330A, 2330B, 2420A.
NM104		General default for NM1(04) segment element values. Loops 2010BA, 2010CA, 2310A, 2310B, 2310CDF, 2330A, 2420A.
NM109		General default for NM1(09) segment element values. Loops 1000A, 1000B, 2010BA, 2310A, 2310CDF, 2330A, 2330B.
N301		General default for N3(01) segment element values. All instances.
N401		General default for N4(01) segment element values. All instances.
N402		General default for N4(02) segment element values. All instances.
N403		General default for N4(03) segment element values. All instances.
PER04		General default for PER(04) segment element values. Loop 1000A, 2010AA
DMG02		General default for DMG(02) segment element values. Loop 2000CA, 2010BA.

Table 9. ImpliedDecimalON

Variable	Default	Description
BuildLog	FALSE	TRUE/FALSE value. Adds RRS logging when building EDI if the ImpliedDecimal setting is enabled. See the General settings.

Table 10. 2000B

Vari able	D e f a u l t	Description
Sub scri berI sPat ient	F A L S E	TRUE/FALSE. Setting for forcing EDI to output Subscriber is the Patient. Default is false.
SBR 03		Sets a default value for SBR03 to be used if the 11IPolNo field is empty.
NO_ SBR 03	F A L S E	When set to TRUE SBR03 will always map as empty. When this variable is set to TRUE and the SBR04 variable is used, then if the if 11cIPlan field is blank, the value for SBR03 will be mapped to the Header section's PayerIDNumber variable.
SBR 04		Sets a default value for SBR04 to be used if the 11cIPlan field is empty.
SBR 09	Z Z	Setting for Loop 2000B SBR(09) claim filing indicator code. Standard values are 11 to 17, AM, BL, CH, CI, DS, FI, HM, LM, MA, MB, MC, OF, TV, VA, WC, and ZZ. This setting is FindValue enabled, so if another value is placed in this setting on the current page, then the document will be searched for a field or variable with this value. The search order is: <ul style="list-style-type: none"> 1. Page Field 2. Page Variable 3. Document Variable <p>If no DCO nodes with this id are found, the value is used.</p>

Table 11. 2010AA

Variabl e	D e f a u l t	Description
BOXPat tern	T R U E	Setting this attribute to TRUE enables detection using a regular expression for <i>PO Box</i> values in the loop 2010AA street address field. If a PO Box is detected, the field value is set to the DefaultStreetAddress setting value by default.
Remap BOX2P ayTo	F A L S E	When this variable is set to TRUE, if a POBox is detected in the field 32 address line, and the field 31 address also has a value, then the field 32 address is remapped to loop 2010AB and the field 31 address is mapped to loop 2010AA.
Default StreetA ddress		Value to replace in the street address field if a PO Box is detected in the address field.

Table 12. 2010BB_BC

Variable	Default	Description
PrimaryPayerName		Setting to use for all processed claims for Loop 2010BB NM1(03). Takes priority over captured value as well as default value set in Header PayerName section.
PrimaryPayerIDNumber		Setting to use for all processed claims for Loop 2010BB NM1(09). Takes priority over default value set in the Header section's PayerIDNumber variable.
PrimaryPayerAddress		Value to place in Loop 2010BB N3(01).
PrimaryPayerCity		Value to place in Loop 2010BB N4(01).
PrimaryPayerState		Value to place in Loop 2010BB N4(02).
PrimaryPayerZip		Value to place in Loop 2010BB N4(03).

Table 13. 2300

Variable	Default	Description
CalculateTotal	TRUE	TRUE/FALSE. Option to calculate the lineitem total using service line data instead of the field 28 total charges field for Loop 2300 CLM(02).
REF D9		Value for Loop 2300 Ref D9 segment. By default, this value is blank, and no Ref D9 is exported. This setting is FindValue enabled, so if another value is placed in this setting on the current page, then the document will be searched for a field or variable with this value. The search order is: <ol style="list-style-type: none"> 1. Page Field 2. Page Variable 3. Document Variable <p>If no DCO nodes with this id are found, the value is used.</p>
REF EA	The <i>dvDCN</i> document variable	Value for Loop 2300 Ref EA segment. This setting is FindValue enabled, so if another value is placed in this setting on the current page, then the document will be searched for a field or variable with this value. The search order is: <ol style="list-style-type: none"> 1. Page Field 2. Page Variable 3. Document Variable <p>If no DCO nodes with this id are found, the value is used.</p>

Table 14. 2310A

Variable	Default	Description
----------	---------	-------------

Variable	Default	Description
AlwaysInclude	FALSE	Setting this value to TRUE will always export Loop 2310A even if fields 23 or 19 do not have a captured value.

Table 15. 2320

Variable	Default	Description
SearchOrder	Z	<p>Value for the Loop 2320 SPR(09) claim filing indicator code. Standard values are 11 to 17, AM, BL, CH, CI, DS, FI, HM, LM, MA, MB, MC, OF, TV, VA, WC, and ZZ. This setting is FindValue enabled, so if another value is placed in this setting on the current page, then the document will be searched for a field or variable with this value. The search order is:</p> <ol style="list-style-type: none"> 1. Page Field 2. Page Variable 3. Document Variable <p>If no DCO nodes with this id are found, the value is used.</p>

Table 16. Loop2320

Variable	Default	Description
OI03	Y	Value for Loop 2320 OI(03).
OI04		Value for Loop 2320 OI(04).
OI06	Y	Value for Loop 2320 OI(06).

Table 17. 2330B

Variable	Default	Description
OtherPayerAddress		Value to place in Loop 2330B N3(01)
OtherPayerCity		Value to place in Loop 2330B N4(01).
OtherPayerState		Value to place in Loop 2330B N4(02).
OtherPayerZip		Value to place in Loop 2330B N4(03).

Table 18. 2400

Variable	Default	Description
AutoPopulateUnit	TRUE	When set to TRUE, this variable will auto populate SV1(03) with MJ and SV1(04) with the service line INFO field if the INFO field has a captured value. Otherwise SV1(03) uses UN and SV1(04) is filled with the Days field value.

Variable	Default	Description
RoundCharges	TRUE	When set to TRUE, this variable toggles rounding of SV1(04) to a whole number.

Table 19. N3

Variable	Default	Description
POBoxDetect	FALSE	When set to TRUE this variable will enable evaluation of all N3 segments for POBox values, and will remap the POBox to the second address line in the N3 segment if the second line is blank.
BuildLog	FALSE	When set to TRUE, this variable will add more detailed RRS logging when creating N3 segments.
KeepTrailingSeparators	FALSE	When set to TRUE, this variable will not remove blank trailing elements from N3 segments.

Table 20. N4

Variable	Default	Description
BuildLog	FALSE	TRUE/FALSE value. Adds more detailed RRS logging when creating N4 segments.
KeepTrailingSeparators	FALSE	TRUE/FALSE value. Option to not remove blank trailing elements from N4 segments.

Table 21. REF

Variable	Default	Description
BuildLog	FALSE	TRUE/FALSE value. Adds more detailed RRS logging when creating REF segments.

Parent topic: [Medical Claims 5010 form configuration parameters](#)

Action library summaries

Information is available for all actions from within Datacap Studio. To access the embedded help, select an action on the Actions Library tab and click the  button.

Descriptions of the available actions are provided in the sections that follow.

- [Global actions](#)
You can use the global actions with any of the Datacap applications.
- [Application specific actions](#)
The Datacap applications use actions that are specific to them.

Global actions

You can use the global actions with any of the Datacap applications.

- [Datacap supported image types](#)
Datacap supports different input files and formats as an image input for the recognition and other different tasks.
- [Autodoc actions](#)
Use the Autodoc actions to create fingerprints and match pages to fingerprints by using the local fingerprint service or a web fingerprint service.
- [Barcode_P actions](#)
Use the Barcode_P actions to locate and read PDF-417 barcodes.
- [ClassifyLayout actions](#)
Use the ClassifyLayout actions to identify page types based on page layout templates.
- [CC actions](#)
Use the CC actions to identify types of pages and other text with IBM® Content Classification technology.
- [Cco2cco actions](#)
Use the Cco2cco actions to sort and filter the words and lines in a fingerprint CCO file.
- [CheckProcessing actions](#)
Use the Check Processing actions to extract data from checks originating from Argentina, Brazil, Canada, France, India, UK and US.
- [CMISClient actions](#)
Content Management Interoperability Services (IBM CMIS) is an open standard that CMISClient actions use to enable communication between Datacap applications and content management systems over the internet.
- [ColorToBW actions](#)
Use the ColorToBW actions to change the color depth of an image.
- [Convert actions](#)
Use the Convert actions to convert various electronic document files into TIFF image files.
- [DatacapBOX actions](#)
Use the DatacapBOX actions to move files between your Datacap system and Box.com.
- [Dcclip actions](#)
Use the Dcclip action to clip a portion of each page image and save it as a separate TIFF file.
- [DCImageFix actions](#)
Use the DCImageFix actions to clean up and enhance page images.
- [DCO actions](#)
Use the DCO actions to set up and modify the runtime batch hierarchy (runtime DCO) information.
- [dcpdf actions](#)
Use the dcpdf actions to convert PDF files to TIFF at the start of the workflow. You can also convert the TIFF files in a document into a PDF file.
- [DocumentAnalytics actions](#)
Use the DocumentAnalytics actions to identify the content type of text blocks and extract data elements from the page layout file.
- [DocumentAnalytics.VisualRecognitionClassifier actions](#)
Visual Recognition Classifier is an IBM Watson service, which can be trained to classify image-based documents. The default instance name: DocumentAnalytics.Undoable-in-transactional.
- [DocumentAnalytics.NaturalLanguageClassifier actions](#)
Natural Language Classifier is an IBM Watson service, which can be trained to classify documents or sections of documents. That based on the text that is contained in the document or the section of a document. The default instance name: DocumentAnalytics.NaturalLanguageClassifierActions.
- [Documentum actions](#)
Use the Documentum actions to upload documents to an EMC Documentum repository.
- [Email actions](#)
Use the email actions to compose and then send an email by using CDOSYS and an SMTP server. These actions also support Outlook, which requires the Outlook user to be logged on to the computer and security prompts might be displayed for each message.

- [Equalize actions](#)
Use the Equalize action to equalize the x and y resolutions of an image.
- [Ewsml actions](#)
Use the Ewsml actions to import image file attachments from an Exchange Server into the current batch by using Exchange Web Service (EWS).
- [Export actions](#)
Use the Export actions to set up and write information to the export text file.
- [ExportDB actions](#)
Use the ExportDB actions to set up and write information to an export database. You build the record in memory before you commit it to the database by using AddRecord.
- [ExportXML actions](#)
Use the ExportXML actions to set up and write information to an export XML file.
- [FileIO actions](#)
Use the FileIO actions to do various file system functions.
- [FileNetIDM actions](#)
Use the FileNetIDM actions to upload documents into an FileNet® Image Services library
- [FileNet P8 actions](#)
Use the FileNet P8 actions to export data to a FileNet P8 repository.
- [FingerprintMaintenance actions](#)
Use the FingerprintMaintenance actions to delete fingerprints from the fingerprint library of the application.
- [FPXML actions](#)
Use the FPXML actions to store zone coordinates in an external XML file instead of the document hierarchy (setup DCO). These actions are useful for porting fingerprints between systems or to avoid making frequent modifications to the document hierarchy.
- [Grayscale actions](#)
Use the Grayscale action to convert grayscale TIFF images to black-and-white.
- [HandwritingRecognition actions](#)
Use the Handwriting Recognition actions to perform cursive recognition of the configured fields using the Parascript FormXtra Recognition Engine.
- [IBMCM actions](#)
Use the IBMCM actions to upload documents into an IBM Content Manager Connector repository.
- [ICR_C actions](#)
Use the ICR_C actions to recognize constrained (unconnected) hand or computer printed characters. These actions use the OpenText RecoStar engine.
- [ImageConvert actions](#)
Use the ImageConvert actions to combine image files or to convert image files to JPEG or TIFF.
- [ImageFix actions](#)
The ImageFix actions are older versions of the DCImageFix action. Use the DCImageFix actions instead.
- [Email actions](#)
Use the Email actions to import image attachments from a mail server into the current batch by using IMAP.
- [Imprint actions](#)
Use the Imprint actions to imprint text over an image, or for blackout or whiteout redactions.
- [Intellocate actions](#)
Use the Intellocate actions to update the existing field position information in the document hierarchy (setup DCO) or to add position information for a new fingerprint.
- [Invoice actions](#)
Use the Invoice actions to process invoices by using the Datacap Accounts Payable application.
- [IOverlay actions](#)
Use the IOverlay actions to combine the current page image with a background image. You can use this action to reapply a form background that dropped out during scanning.

- [Locate actions](#)
Use the Locate actions in combination with full text recognition to locate words or regular expressions on the page. And to move around the page by line or word.
- [Lookup actions](#)
Use the Lookup actions to validate field values by using database lookups and populate fields with lookup results.
- [MC_Identify](#)
Use the MC_Identify actions to identify claim forms in a batch.
- [MC_Validation](#)
Use the MC_Validation actions to validate medical claim form information.
- [mvscan actions](#)
Use mvscan actions in place of the vscan actions to create batches from files that are stored on your local disk. The mvscan actions can handle large numbers of files that are stored in the input folder.
- [Maintenance Manager actions](#)
The Maintenance Manager actions are divided into setup, batch processing, logging, and reporting categories.
- [OCR_J actions](#)
The OCR_J actions use the Rosetta Stone Japanese OCR engine to perform OCR for Japanese machine print and Japanese handwriting.
- [OCR_A actions](#)
Use the OCR_A actions to do text recognition by using the FineReader OCR engine.
- [OCR_N actions](#)
Use the OCR_N actions to do recognition by using the NovoDynamics engine. The OCR_N actions can run recognition on a full page or on all of the field zones that are defined for the current page.
- [OCR_SR actions](#)
Use the OCR_SR actions to do recognition and image rotation by using the updated Nuance OmniPage OCR engine. You can run recognition actions on field zones and pages. The results can be processed by actions in other libraries and ultimately displayed to the user for verification. The results can also be written out to files in several possible file formats.
- [OpenTextFaxServer actions](#)
Use the OpenTextFaxServer actions to import faxes from an OpenTextFaxServer.
- [PatternMatch actions](#)
Use the PatternMatch actions for pattern-based page identification and for page registration (alignment). Page registration is important when you are working with OMR check boxes.
- [Picture actions](#)
Use the Picture actions to do field validations by picture strings. Picture strings define the supported format of a field such as a social security number, phone number, date.
- [POLR actions](#)
Use the POLR action matches line items from your invoice image to the corresponding purchase order.
- [Recog_Shared actions](#)
Use the Recog_Shared actions to do various fingerprint and recognition-related functions.
- [runner actions](#)
Use the runner actions to do miscellaneous utility functions.
- [SharedRecognitionTools actions](#)
Use the actions in this library after you perform OCR. For example, you might perform OCR with actions from the OCR_A or OCR_SR libraries and then use the actions in this library. Actions that can produce the layout XML include OCR_SR.Recognize and OCR_A.Recognize, both of which can process color images and PDF files. To use the Locate actions and perform click 'n' key during verification, use the action CreateCcoFromLayout to create a CCO file for the page after producing the layout XML file.
- [SignatureValidation actions](#)
Use the Signature Validation actions to detect and validate signatures on a document or zone.
- [SPEXport actions](#)
Use the SPEXport actions to upload documents to a Microsoft SharePoint library.

- [Split actions](#)
Use the Split action to split a batch into smaller batches so each can be processed separately.
- [Statistics actions](#)
These statistics actions are used by the Profile Statistics and Export Statistics rulesets to save information about field recognition accuracy and page classification accuracy.
- [TifMerge actions](#)
Use the TifMerge actions to combine individual TIFF images into a multi-page TIFF file. This action is typically run at the end of the workflow so that you can upload or release the batch images as a single file.
- [TM524 actions](#)
The TM524 actions are for compatibility with older versions of Datacap and are no longer used
- [Validations actions](#)
Use the Validations actions to check and modify the content and format of the current field value.
- [Vote actions](#)
Use the Vote action when you do multi-pass data entry to check whether the first and second passes match.
- [Vscan actions](#)
Use the Vscan actions to create a batch by using existing image files.
- [Web Services actions](#)
This library of actions facilitates communication with external web services. You can use these actions to retrieve formatted responses or image files. The actions can handle both XML and JSON formatted responses, with XML as the default choice. You will need to know the web service's endpoint, any applicable parameters, and the response type before using these actions.
- [Zones actions](#)
Use the Zones actions to work with the zones that define the position of each field on the page.

Parent topic: [Action library summaries](#)

Datacap supported image types

Datacap supports different input files and formats as an image input for the recognition and other different tasks.

When you perform recognition of text or barcodes, it is recommended to use a file format that supports lossless compression. Lossy compression, such as JPEG compression, is intended for photographic content, not textual content. Lossy compression can cause line edges to become soft or blurred, which can degrade characters and barcodes, causing recognition errors.

The following is the list of image types which Datacap supports:

Action Library	Action/Feature	Input Image Type	Color Depth	Compression	Notes
Barcode P	All actions	TIF, BMP, JPEG, PNG	Up to 24 bit	TIF - Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	

Action Library	Action/Feature	Input Image Type	Color Depth	Compression	Notes
ColorToBW	C2BW_Convert	TIF, BMP, JPEG, PNG	Up to 24 bit	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	
Convert.Images	ImageToTIFF	BMP, JPEG, PNG (non-interlaced)	Up to 24 bit	Uncompressed, Standard	
Dcclip	dci_clipfield	TIF		CCITT Group 3 & 4, Huffman encoding, LZW, Packbits, Uncompressed	
Grayscale	ConvertGrayToBW	TIF	8 bpp	Huffman encoding, LZW, Packbits, Uncompressed	
ICR_C	All image input actions	TIF	1 bpp	Fax, Huffman, LZW, Uncompressed	
CheckProcessing	ProcessCheck	JPEG	Depends on country	N/A	
CheckProcessing	ProcessCheck	TIF	Depends on country	CCITT Group 3 & 4, Huffman encoding, LZW, Packbits, Uncompressed	
Image Enhancement	Adjust Brightness/Contrast	TIF, JPEG, BMP	8 gray, 24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 1, 8 bpp color.

Action Library	Action/Feature	Input Image Type	Color Depth	Compression	Notes
Image Enhancement	Auto Rotate	TIF, BMP	1 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	It does not support 8 color, 8 gray, 24 bpp.
Image Enhancement	Binaries	TIF, JPEG, BMP	1, 8 color, 8 gray, 24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	
Image Enhancement	Border Removal	TIF, JPEG, BMP	1, 8 gray, 24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 8 color.
Image Enhancement	Close Filter	TIF, JPEG, BMP	8 gray, 24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 1, 8 color bpp.

Action Library	Action/Feature	Input Image Type	Color Depth	Compression	Notes
Image Enhancement	Deskew	TIF, JPEG, BMP	1, 8 gray, 24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 8 color.
Image Enhancement	Despeckle	TIF, BMP	1 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 8 color, 8 gray, 24 bpp.
Image Enhancement	Dilate	TIF, JPEG, BMP	1, 8 gray, 24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 8 color.
Image Enhancement	Erode	TIF, JPEG, BMP	1, 8 gray, 24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 8 color.

Action Library	Action/Feature	Input Image Type	Color Depth	Compression	Notes
Image Enhancement	Flip	TIF, JPEG, BMP	1, 8 gray, 24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 8 color.
Image Enhancement	Image Detergent	TIF, JPEG, BMP	24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 1, 8 color, 8 gray bpp.
Image Enhancement	Image Registration	TIF, JPEG, BMP	1, 8 gray, 24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 8 color.
Image Enhancement	Inverse Text Correction	TIF, BMP	1 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	It does not support 8 color, 8 gray, 24 bpp.

Action Library	Action/Feature	Input Image Type	Color Depth	Compression	Notes
Image Enhancement	Mirror	TIF, JPEG, BMP	1, 8 gray, 24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 8 color.
Image Enhancement	Remove Blobs	TIF, BMP	1 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	It does not support 8 color, 8 gray, 24 bpp.
Image Enhancement	Remove Combs	TIF, BMP	1 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	It does not support 8 color, 8 gray, 24 bpp.
Image Enhancement	Remove Dot Shading	TIF, BMP	1 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	It does not support 8 color, 8 gray, 24 bpp.
Image Enhancement	Remove Hole Punch	TIF, BMP	1 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	It does not support 8 color, 8 gray, 24 bpp.

Action Library	Action/Feature	Input Image Type	Color Depth	Compression	Notes
Image Enhancement	Remove Lines	TIF, BMP	1 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	It does not support 8 color, 8 gray, 24 bpp.
Image Enhancement	Rotate Image	TIF, JPEG, BMP	1, 8 color, 8 gray	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	
Image Enhancement	Smooth Background	TIF, JPEG, BMP	8 gray, 24 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed JPEG - Standard Encoding	It does not support 1, 8 color, 8 gray bpp.
Image Enhancement	Smooth Objects	TIF, BMP	1 bpp	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	It does not support 8 color, 8 gray, 24 bpp.
ImageConvert	AppendAllImages	TIF	1 bpp, 8 bpp grayscale, 24 bpp	TIF - CCITT Group 3 & 4, Huffman encoding, LZW, Packbits, Uncompressed	It does not support 8 bpp color.
ImageConvert	AppendAllImages_ByType	TIF	1 bpp, 8 bpp grayscale, 24 bpp	CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed	It does not support 8 bpp color.

Action Library	Action/Feature	Input Image Type	Color Depth	Compression	Notes
ImageConvert	AppendImage	TIF	1 bpp, 8 bpp grayscale, 24 bpp	CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed	It does not support 8 bpp color.
ImageConvert	AppendImage_StartAsNew	TIF	1 bpp, 8 bpp grayscale, 24 bpp	CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed	It does not support 8 bpp color.
ImageConvert	ConvertToJPEG	TIF, BMP, PNG	Up to 24 bit	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	
ImageConvert	ConvertToTIFF	BMP, JPEG, PNG	Up to 24 bit	Uncompressed, Standard	
ImageConvert	RescaleImage	TIF, BMP, JPEG, PNG	Up to 24 bit	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	
ImageConvert	SetImageDPIByWidth	TIF, BMP, JPEG, PNG	Up to 24 bit	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	
OCR_A	All image input actions	TIF, BMP, JPEG, PNG	Up to 24 bit	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	

Action Library	Action/Feature	Input Image Type	Color Depth	Compression	Notes
OCR_SR	All impage input actions	TIF, BMP, JPEG, PNG	Up to 24 bit	TIF - CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed BMP - Uncompressed	
TifMerge	TifMerge_Merge Images	TIF		CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed	
TifMerge	TifMerge_MyImage	TIF		CCITT Group 3 & 4, Huffman, LZW, Packbits, Uncompressed	

Parent topic: [Global actions](#)

Autodoc actions

Use the Autodoc actions to create fingerprints and match pages to fingerprints by using the local fingerprint service or a web fingerprint service.

The Autodoc actions connect you to a fingerprint service. You run these actions to specify how to create fingerprints, set up fingerprint matching processes, and update fingerprint statistics.

- [BlankPagesIDBySize](#)
Uses the size of the Image file to determine if the file represents a blank page.
- [CalculateOffset](#)
Sets the standard Offset value to be used when matching source pages to fingerprints.
- [CreateFingerprint](#)
Adds the image (TIF) file of the current page and the fingerprint (CCO) file to the fingerprint library of the application. The resulting fingerprint consists of the Image file (.tif) of the page and its Processing file (.cco).
- [DeleteFingerprint](#)
Removes the image (TIF) file of the current page and the fingerprint (CCO) file from the fingerprint library of the application.
- [FindBlackFingerprint](#)
Attempts to match black forms to fingerprints in the fingerprints directory of an application. If a match does not occur, the action responds according to the parameters that you enter.
- [FindFingerprint](#)
Attempts to match the current page to a fingerprint and creates a new fingerprint if a match does not occur.
- [MergeCCOs_ByType](#)
Merges the fingerprint files (.cco) that are associated with Page objects of the Document Hierarchy.
- [MergeLayoutByType](#)
This action merges the layout files (*layout.xml) associated with Page objects of the Document Hierarchy.

- [SetApplicationID](#)
Used with the Fingerprint Service to limit a fingerprint search to the specified application.
- [SetFilter_HostName](#)
Limits a fingerprint match to the specified fingerprint class only.
- [SetFilter_PageType](#)
Limits a fingerprint match to the specified page type only.
- [SetFingerprint](#)
Sets the class and page type after you create a new fingerprint.
- [SetFingerprintDir](#)
Specifies the Fingerprint directory of your application.
- [SetFingerprintFailureThreshold](#)
Specifies the percentage of fingerprint upload failures to ignore when you use the Fingerprint web service.
- [SetFingerprintSearchArea](#)
Specifies the portion of the current page that is used during fingerprint matching.
- [SetFingerprintWebServiceURL](#)
Specifies the URL that links to the Fingerprint web service.
- [SetMaxOffset](#)
Sets the maximum offset value to use while matching source pages to fingerprints.
- [SetProblemValue](#)
Uses the decimal value that you supply as a parameter to set a minimum Matching Tolerance Rating.
- [UpdateFingerprintStats](#)
Updates the fingerprint statistics in the Fingerprint database.

Parent topic: [Global actions](#)

BlankPagesIDBySize

Uses the size of the Image file to determine if the file represents a blank page.

Syntax

```
bool BlankPagesIDBySize (StrParam)
```

Parameters

A three-part, comma-separated value consisting of:

1. Numeric value indicating the maximum size in bytes that qualifies a page as a blank page.
2. String value representing the Page Type of a blank page.
3. Numeric value (0, 1 or 2) to designate which pages in a multi-page Image file are to be evaluated.

The third parameter is optional.

- 0 = both sides of a two-page Image file.
- 1 = odd pages only.
- 2 = even images only.

»

Returns

False if any of the following conditions apply:

- The parameter is invalid
- The rule with this action is bound to a field object of the document hierarchy
- Any child image of the calling object qualifies as a blank page based on the specified byte size

Otherwise, this action returns True.◀

Level

Batch, Document, or Page levels.

Details

Any page with an Image file smaller than the size parameter (in bytes) is assigned the Page Type value you enter as a parameter.

Example

```
BlankPagesIDBySize ("1000,Blank_Page")
```

Parent topic: [Autodoc actions](#)

CalculateOffset

Sets the standard Offset value to be used when matching source pages to fingerprints.

Syntax

```
bool CalculateOffset ()
```

Returns

False if the action is not applied at the Page level or an error occurs. Otherwise, True.

Level

Page level only.

Details

Calculates the Confidence and Image_Offset variables (similarity with the Fingerprint and image shift compared with the Fingerprint image) for the page based on its relation to the page's fingerprint. The fingerprint ID must previously be set using Findfingerprint, SetFingerprint or some other method.

Example

```
CalculateOffset ()
```

Parent topic: [Autodoc actions](#)

CreateFingerprint

Adds the image (TIF) file of the current page and the fingerprint (CCO) file to the fingerprint library of the application. The resulting fingerprint consists of the Image file (.tif) of the page and its Processing file (.cco).

Syntax

```
bool CreateFingerprint()
```

Parameters

None.

Returns

False if the rule with this action is not bound to a Page object of the Document Hierarchy; if the current page does not have an Image file; or if the fingerprint's two files cannot be created. Otherwise, True.

Level

Page level only.

Details

Creates a fingerprint for the current source page. The resulting fingerprint will consist of two files: the page's Image file (.tif) and its Processing file (.cco).

Attention: A SetFingerprintDir action must precede this action.

Example

```
SetFingerprintDir("C:\ParentDirectory\Application\fingerprint")  
CreateFingerprint() CreateFingerprint()
```

Parent topic: [Autodoc actions](#)

Related reference:

[SetFingerprintDir](#)

[DeleteFingerprint](#)

DeleteFingerprint

Removes the image (TIF) file of the current page and the fingerprint (CCO) file from the fingerprint library of the application.

Syntax

```
bool DeleteFingerprint()
```

Parameters

None.

Returns

Always returns True. Under certain conditions the action will be unable to delete the fingerprint but will still return True. For example if the action is not applied at the Page level or if the fingerprint's Image file cannot be found. Please review the log file if DeleteFingerprint does not perform as expected.

Level

Page level only.

Details

Deletes the Image file (.tif) And Processing file (.cco) of the current page's fingerprint from the application's fingerprint directory, and its record from the Fingerprint database.

Attention: A SetFingerprintDir action must precede this action.

Example

```
SetFingerprintDir ("C:\ParentDirectory\Application\fingerprint")
DeleteFingerprint ()
```

Parent topic: [Autodoc actions](#)

Related reference:

[SetFingerprintDir](#)

[CreateFingerprint](#)

FindBlackFingerprint

Attempts to match black forms to fingerprints in the fingerprints directory of an application. If a match does not occur, the action responds according to the parameters that you enter.

Syntax

```
bool FindBlackFingerprint (StrParam)
```

Parameters

Two comma-separated values (the second is optional)

1. True or False: True If a fingerprint match is not found, a new fingerprint is created and the two fingerprint files (.tif and .cco) are placed in the fingerprint directory. False, if the task is to proceed without creating a new fingerprint.
2. Optional: The Page Type that is to be assigned to the newly created fingerprint if the first parameter is True. If you do not include this parameter, the action assigns the Page Type of the current page.

Returns

False, if the action is not applied at the Page level or if the first parameter is False and a fingerprint match does not occur. Otherwise, True.

Level

Page level only.

Details

This action attempts to match black forms to fingerprints in the fingerprints directory of an application.

If a match does not occur, the action responds according to the parameters you enter.

Example:

```
AnalyzeImage ()
SetSearchArea ("0.5")
SetProblemValue ("0.7")
SetFingerprintDir ("ParentDirectory\Application\fingerprint")
FindBlackFingerprint ("True, PageType")
```

In this sequence, the FindBlackFingerprint action uses only the first 50% of the fingerprint to search for a match. It accepts a match of 0.7 or higher.

If no match is found, the sequence creates a new fingerprint and stores it in the location that is specified by the SetFingerprintDir action.

If the parameter is set to False, no fingerprint is created when a match is not found.

Parent topic: [Autodoc actions](#)

FindFingerprint

Attempts to match the current page to a fingerprint and creates a new fingerprint if a match does not occur.

Syntax

```
bool FindFingerprint (StrParam)
```

Parameters

Two comma-separated values (the second is optional).

1. A True / False value: True, if a task is to create a new fingerprint and add it to the fingerprint directory when a match does not occur; False, if the task is to proceed without creating a new fingerprint.
2. Optional: The Page Type that is to be assigned to the newly created fingerprint. If omitted, the current Page Type of the current page is used.

Returns

False if the action is not applied at the Page level, or if the parameter is False and a fingerprint match does not occur. Otherwise, True. If a new Fingerprint cannot be added, the action still returns True.

Level

Page level only.

Details

This action attempts to match the current source page to a fingerprint, and creates a new fingerprint if a match does not occur. Include this action after a rule's SetSearchArea, SetProblemValue, and SetFingerprintDir actions.

Example

```
AnalyzeImage ()
RotateImage ()
RecognizePageOCR_S ()
SetSearchArea ("0.5")
SetProblemValue ("0.7")
```

```
SetFingerprintDir ("\\ParentDirectory\\Application\\Fingerprint")
FindFingerprint ("True, Invoice_Page")
```

In this example sequence, the FindFingerprint action uses only the first 50% of the current page to search for a match. However, it accepts a match of 0.7 or higher. If no match is found, the sequence creates a new fingerprint and stores it in the location that is specified by the SetFingerprintDir action. If the parameter is set to False, a new fingerprint is not created when there is no match.

Parent topic: [Autodoc actions](#)

MergeCCOs_ByType

Merges the fingerprint files (.cco) that are associated with Page objects of the Document Hierarchy.

Syntax

```
bool MergeCCOs_ByType (StrParam)
```

Parameters

Comma-separated String values that indicate the Page Types of the Document Hierarchy objects to be merged.

Returns

False, if a Fingerprint file (.cco) for one of the Page Types is not available. Otherwise, True.

Level

Document level only.

Details

Merges the Fingerprint files (.cco) associated with Page objects of the Document Hierarchy.

Example

```
MergeCCOs_ByType ("Invoice_Page, Invoice_Cont")
```

The MergeCCOs_ByType action allows all values of the source pages to be assigned to a single, searchable Fingerprint Processing file (.cco).

Parent topic: [Autodoc actions](#)

MergeLayoutByType

This action merges the layout files (*layout.xml) associated with Page objects of the Document Hierarchy.

Syntax

```
bool MergeLayoutByType (string PageTypesToMerge)
```

Parameters

Comma-separated string values that indicate the Page Types of the Document Hierarchy objects to be merged. Smart parameters are supported.

Returns

False, if a layout file (*layout.xml) for one of the Page Types is not available or if the merged layout cannot be created. Otherwise, True.

Level

Document level only.

Details

The layout.xml for the first page that is merged becomes the new merged layout.xml. The merged layout.xml can be then loaded to create a merged CCO for the merged pages. The layout file for the first merged page is updated to contain the recognition results from the layout block information for all of the merged pages.

Only pages that are contained in the current document object are merged. A layout file must exist for each of the pages to be merged. Only page types that are indicated by the parameter are included in the merged file.

Example

```
MergeLayoutByType ("Main_Page,Trailing_Page")
```

The MergeLayoutByType action combines all of the text of the pages in the current DCO document of type Main_Page and Trailing_Page into a single, searchable layout file.

Parent topic: [Autodoc actions](#)

SetApplicationID

Used with the Fingerprint Service to limit a fingerprint search to the specified application.

Syntax

```
bool SetApplicationID(StrParam)
```

Parameters

String parameter that represents unique Application ID. Smart parameters are supported.

This value is used to retrieve a correct list of fingerprints that are loaded to the server.

Returns

False, if action SetFingerprintWebServiceURL() was not called. Otherwise, True.

Level

Batch, Document, or Page levels.

Details

Uses this action to specify unique application name, if multiple applications are using the same Fingerprint Service.

Example

```
SetFingerprintWebServiceURL (http://'FPSEVERNAME'/fpsevice/Service.asmx?WSDL)
SetApplicationID("1040ez")
```

Parent topic: [Autodoc actions](#)

SetFilter_HostName

Limits a fingerprint match to the specified fingerprint class only.

Syntax

```
bool SetFilter_HostName (StrParam)
```

Parameters

The String value of the Fingerprint Class you want to use as the filter.

Returns

Always True.

Level

Page level.

Details

Sets the Fingerprint Class filter for the identification (matching) algorithm. Smart Parameters are supported.

The filter forces the fingerprint-matching algorithm to use only fingerprints associated with the specified Fingerprint Class.

To disable this filter, call with an empty string as the parameter.

Example

```
SetFilter_HostName ("MyFingerprintClass")
```

Parent topic: [Autodoc actions](#)

SetFilter_PageType

Limits a fingerprint match to the specified page type only.

Syntax

```
bool SetFilter_PageType (StrParam)
```

Parameters

The String value of the Page Type you want to use as the filter.

Returns

Always True.

Level

All levels, but generally at the Page level.

Details

Sets the Page Type Filter For the identification (matching) algorithm. Smart Parameters are supported.

The filter will force the fingerprint-matching algorithm to use only fingerprints associated with that Page Type.

Example

```
SetFilter_PageType ("Invoice_Page")
```

Parent topic: [Autodoc actions](#)

SetFingerprint

Sets the class and page type after you create a new fingerprint.

Syntax

```
bool SetFingerprint (StrParam)
```

Parameters

A two-part, comma-separated value consisting of:

1. The Fingerprint Class value. Smart parameters are supported. Alternatively, the name of the Field on this page that specifies the Fingerprint Class.
2. (optional) The Fingerprint ID value. Smart parameters are supported. Alternatively, the name of the Field on this page that specifies the Fingerprint ID.

Returns

False if either parameter is invalid. Otherwise, True.

Level

Page level only.

Details

Sets a newly created fingerprint's Fingerprint Class and Fingerprint Class ID values.

After the mandatory Fingerprint Class value and optional Fingerprint Class ID have been manually assigned by a fingerprint creation task, this action places these values into the Host table of the application's Fingerprint database - as RefName and RefID values.

Example

```
SetFingerprint("@P\VendorName,@P\VendorID")
```

In this example, runtime values of the VendorName and VendorID Field objects will populate the Host table of the application's Fingerprint Database. Alternative method:

```
SetFingerprint("@VendorName,@VendorID")
```

Parent topic: [Autodoc actions](#)

SetFingerprintDir

Specifies the Fingerprint directory of your application.

Syntax

```
bool SetFingerprintDir(StrParam)
```

Parameters

A String value that specifies the directory's name and path. Smart parameters are supported. A Drive ID, such as C:\, is optional.

Returns

Always True.

Level

All levels, but usually applied at the Page level.

Details

Sets the Fingerprint directory of your application. This directory contains the application's fingerprints.

Example

```
SetFingerprintDir("C:\ParentDirectory\Application\Fingerprint")
```

This action identifies the location and path of an application's Fingerprint directory.

Parent topic: [Autodoc actions](#)

SetFingerprintFailureThreshold

Specifies the percentage of fingerprint upload failures to ignore when you use the Fingerprint web service.

Syntax

```
bool SetFingerprintFailureThreshold(StrParam)
```

Parameters

Int parameter that represents percent threshold of fingerprint upload failures to ignore. Smart parameters are supported.

The batch aborts if the percentage of fingerprints that are failed to load exceeds this value.

Returns

Returns False when:

- The fingerprint service is not configured using action `SetFingerprintWebServiceURL()`.

Returns **False** and aborts the batch when:

- The input parameter is not a value from 0 to 100.
- If the percentage of fingerprints that failed to load is greater than the threshold.
- If no fingerprints have been loaded and at least one has failed to load.

Otherwise, True.

Level

Batch, Document, or Page levels.

Details

Uses this action to set the maximum number of fingerprint upload failures to ignore. If the application has been set using the action `SetApplicationID`, then the threshold will be set for that specific application only.

Example

```
FindFingerprint (True)
SetFingerprintFailureThreshold ("10")
```

Parent topic: [Autodoc actions](#)

SetFingerprintSearchArea

Specifies the portion of the current page that is used during fingerprint matching.

Syntax

```
bool SetFingerprintSearchArea(string matchStart, string matchEnd)
```

Parameters

string matchStart

string matchEnd

Parameters

Two parameters:

1. A decimal value from 0.01 (1%) to 1.00 (all) to indicate how much of the page is to be matched. If the second parameter is empty the first parameter represents the bottom of the area for matching, starting

from the top of the page. If the second parameter is not empty, the first parameter is the top or start of the area for matching. For example, a single parameter of 0.5 indicates that fingerprint matching is limited to the first half of the page (0 - 50%). Decimal separators must be appropriate for the current locale. In applications that might be used in locales with different decimal separators, use percentage notation.

2. Optional: A decimal value from 0.01 (1%) to 1.0 (all) to indicate the end point on the page to be used for fingerprint matching. If this parameter is supplied, the first parameter is the starting point. For example: if the first parameter is 0.6, and the second parameter is 1.0, the last 40% of the page is used for fingerprint matching (60-100%).

Attention: In both cases, you can replace a decimal value with a percentage. To indicate that the value is a percentage, the action can use a number followed by `p`, such as `50p` to represent 50%.

When you are using the percentage values, the number must be a whole number and must not contain a decimal separator. Decimal separators must be appropriate for the current locale. In applications that are used in locales with different decimal separators, use percentage notation.

Returns

False, if the first parameter is missing or is not numeric. Also returns False, if the second parameter is not numeric. Otherwise, True.

Level

All levels, but generally at the Page level.

Details

This action uses the numeric values that you supply to determine the portion of the current page that is used to find a matching fingerprint.

Example

```
SetFingerprintSearchArea ("0.5", "")
```

This example compares lines and words in the upper 50% of the current page to the lines and words in the same portion of each fingerprint. Notice that the second parameter is empty.

```
SetFingerprintSearchArea ("0.5", "1.0")
```

This example compares lines and words in the lower 50% of the current page to the lines and words in the same portion of each fingerprint. You can replace a parameter's decimal value with a percentage(`p`) or metric(`m`) number.

Parent topic: [Autodoc actions](#)

Related reference:
[SetProblemValue](#)

SetFingerprintWebServiceURL

Specifies the URL that links to the Fingerprint web service.

Syntax

```
bool SetFingerprintWebServiceURL (StrParam)
```

Parameters

String value specifying the URL that is the link to the Fingerprint Web Service. Smart parameters are supported.

Returns

Always True.

Level

All levels.

Details

Add `SetFingerprintWebServiceURL` with its single parameter to the function and move so it is prior to `SetFingerprintDir` action.

Attention: this action is not effective unless the Fingerprint Service has been installed and configured.

Example

```
SetFingerprintWebServiceURL("http://www.grandcorp.AR.com/fpservice/")
SetFingerprintDir("\ParentDirectory\Application\Fingerprint")
```

Parent topic: [Autodoc actions](#)

SetMaxOffset

Sets the maximum offset value to use while matching source pages to fingerprints.

Syntax

```
bool SetMaxOffset(StrParam)
```

Parameters

Integer value between 1 and 255.

Returns

False, if the parameter is invalid. Otherwise, True.

Level

All levels.

Details

Sets the Maximum Offset value while matching source pages to fingerprints.

This action has no effect when used with the Fingerprint Service. The actual shift is four times the maximum offset value in pixels ($4 * \text{MaxOffset}$). Increasing this value improves but slows the matching process.

The default value is 6: $4 * 6 = 24$ pixels.

Example

```
SetMaxOffset ("12")
```

This example results in a Maximum Offset value of 48 pixels, which is greater than the 24 pixel default.

Parent topic: [Autodoc actions](#)

SetProblemValue

Uses the decimal value that you supply as a parameter to set a minimum Matching Tolerance Rating.

Syntax

```
bool SetProblemValue (StrParam)
```

Parameters

A decimal value from 0.00 (Lowest Tolerance) to 0.99 (Highest Tolerance). The decimal separator must be appropriate for the locale.

Returns

False, if the parameter is missing or the parameter is not numeric. Otherwise, True.

Level

All, but usually at the Page level.

Details

Uses the decimal value that you supply as a parameter to set a minimum Matching Tolerance Rating.

Important: A lower rating results in lower tolerance and a greater chance for a match, but also a greater chance for a False match.

Example

```
AnalyzeImage ()  
CreateFields ()  
RotateImage ()  
RecognizePageOCR-S ()  
SetSearchArea ("0.5")  
SetProblemValue ("0.70")  
SetFingerprintDir ("\\ParentDirectory\\Application\\Fingerprint")  
FindFingerprint ("True")
```

In this sequence, the FindFingerprint action assigns a Matching Tolerance Rating that is not overly restrictive or unrealistically accepting. If the rule's conditions do not result in a match, and True is used as the parameter for the FindFingerprint action, a new fingerprint is added to the library.

Parent topic: [Autodoc actions](#)

UpdateFingerprintStats

Updates the fingerprint statistics in the Fingerprint database.

Syntax

```
bool UpdateFingerprintStats()
```

Parameters

None.

Returns

False if called from any level other than Page level, or the Fingerprint database is not accessible. Otherwise, True.

Level

Page level.

Details

Use this action to increment the count for the current page and update the fingerprint statistics in the fingerprint database.

Example

```
UpdateFingerprintStats()
```

Parent topic: [Autodoc actions](#)

Barcode_P actions

Use the Barcode_P actions to locate and read PDF-417 barcodes.

The Barcode_P actions recognizes various barcode types, searches the page for all barcodes and writes them to a list, and reads the value of the barcodes.

- [Get2DCodeBP](#)
Recognizes PDF-417 codes.
- [GetAllBarcodesBP](#)
Searches the current page for all barcodes and writes them to the *GetBarCodeList* variable of the calling object.
- [GetBarcodeBP](#)
Recognizes arbitrary 1D or 2D codes.
- [GetDataMatrixCodeBP](#)
Recognizes Data Matrix codes
- [IdentifyByBarcodesBP](#)
Updates the current page type if a barcode match is found.
- [MatchBarcodeBP](#)
Searches all the barcodes on the current page and checks if one of them matches the value that you entered as a parameter.
- [MatchBarcodePrefixBP](#)
Searches all the barcodes on the current page and checks if one of them matches the value that you entered as a parameter.
- [ReadBarCodeBP](#)
Tests to determine if the first barcode on the current page contains the value that is specified by the

- parameter.
- [SetMinimumConfidenceBP](#)
Sets the minimum confidence level that is required for barcodes that are read by the engine to be accepted.

Parent topic: [Global actions](#)

Get2DCodeBP

Recognizes PDF-417 codes.

Member of namespace

Barcode_P

Syntax

```
bool Get2DCodeBP ()
```

Parameters

None.

Returns

True if the action is called at the page level or field level. Otherwise, False.

In addition the calling object's value and variable *GetBarcode* is filled with the bar code value. This action also stores barcode information such as confidence, coordinates, code name, and size. If the object's barcode settings are set to read more than one barcode, and more than one barcode is found, barcodes are also stored in the variable *GetBarcodeX* where X is the index of the barcode found.

Level

Page or Field level only.

Details

Use this action if your page has PDF-417 codes. Any field must have a Position assigned when bar code reading is performed. For example the application would invoke the CreateFields action, and if the field has Position defined in the Document Hierarchy, it is ready for barcode reading. If anchors or fingerprint matching are used, ReadZones or other registration may be required to align the fields correctly.

Example:

```
Get2DCodeBP ()
```

Parent topic: [Barcode_P actions](#)

GetAllBarcodesBP

Searches the current page for all barcodes and writes them to the *GetBarcodeList* variable of the calling object.

Member of namespace

Barcode_P

Syntax

```
bool GetAllBarcodesBP (StrParam)
```

Parameters

The separator to use when storing multiple barcodes. The default separator is a comma.

Returns

True if the action is called at the page level or field level. Otherwise, False.

In addition the calling object's value and variable *GetBarcode* is filled with the bar code value. This action also stores barcode information such as confidence, coordinates, code name, and size. If the barcode settings of the object are set to read more than one barcode, and more than one barcode is found, barcodes are also stored in the variable *GetBarcodeX* where X is the index of the barcode found.

Level

Page and field level.

Details

Searches all the barcodes on the current page and stores them in the *GetBarcodeList* variable of the calling object. Each barcode value is separated using the string separator value entered as a parameter.

If the engine does not detect any barcodes, then the variable *GetBarcodeList* is not populated nor created.

Example:

```
GetAllBarcodesBP (",")
```

Parent topic: [Barcode_P actions](#)

GetBarcodeBP

Recognizes arbitrary 1D or 2D codes.

Member of namespace

Barcode_P

Syntax

```
bool GetBarcodeBP ()
```

Parameters

None.

Returns

True, if the action is called at the page level or field level. Otherwise, False.

In addition, the calling object's value and variable *GetBarcode* is filled with the barcode value. This action also stores barcode information such as confidence, coordinates, code name, and size. If the object's barcode settings are set to read more than one barcode, and more than one barcode is found, barcodes are also stored in the variable *GetBarcodeX* where X is the index of the barcode found.

Level

Page or Field level only.

Details

Use this action if your page has 1D or 2D barcodes. Any field must have a Position assigned when barcode reading is performed. For example, the application starts the CreateFields action, and if the field has Position defined in the Document Hierarchy, it is ready for barcode reading. If anchors or fingerprint matching is used, ReadZones or other registration might be needed to align the fields correctly.

To explicitly tell the engine which barcode types to be read, the field/page being recognized contains a variable or setup property called *bp_tp*. If the *bp_tp* variable is empty or does not exist, the engine defaults to Unknown. See the information for Unknown in the valid barcode types.

The value of this variable must be a combination of the barcode types. This variable is automatically set when you choose the barcode types through the Zones tab in Datacap Studio, under the BAR/P recognition settings.

Multiple types can be read by adding the values of their codes together.

When the barcode type is set to Patch Code, the following string values might be returned by the engine. The string value depends on the type of Patch Code that is found:

- Patch 1 - 1100
- Patch 2 - 1001
- Patch 3 - 1010
- Patch 4 / Toggle Patch - 0110
- Patch 6 - 0011
- Patch T / Transfer patch - 0101

Important: The barcode type must be set explicitly to PatchCode, when you want to recognize PatchCodes. The default barcode type setting is Unknown and it does not detect PatchCodes.

Valid barcode types

- 0 - Unknown. Detects all barcode types automatically, except for PatchCode and PDF417. The PatchCode and PDF417 barcode types must be set explicitly to be detected.
- 1 - INDUSTRY 2 OF 5
- 2 - INTERLEAVED 2 OF 5
- 4 - IATA 2 OF 5
- 8 - DATALOGIC 2 OF 5
- 16 - INVERT 2 OF 5
- 32 - BCD MATRIX
- 64 - MATRIX 2 OF 5
- 128 - CODE 32
- 256 - CODE 39

- 512 - CODABAR 2
- 1024 - CODE 93
- 2048 - CODE 128
- 4096 - EAN-13
- 8192 - EAN-8
- 16384 - UPC-A
- 32768 - UPC-E
- 65536 - ADD 5
- 131072 - ADD 2
- 262144 - UCC128/EAN-128
- 524288 - Patch Code
- 1048576 - PostNet
- 2097152 - PDF417
- 4194304 - DataMatrix
- 8388608 - Code 39 Extended
- 16777216 - Code 93 Extended
- 33554432 - QRCode
- 67108864 - IntelligentMail
- 134217728 - Royal Mail (RM4SCC)
- 268435456 - Australian Post 4-State Code
- 536870912 - Aztec
- 1073741824 - GS1DataBar

Minimum and Maximum Settings

Multiple barcodes can exist and be recognized on a single page. The DCO variable *bp_su* controls the maximum number of barcodes attempts to be read on a single page. If not specified, the default maximum is 10.

The minimum number of expected barcodes can be specified by using the DCO variable *bp_minExpected*. This setting controls when image enhancement takes place. Unless specified, the minimum expected barcodes will default to the same value as the maximum number of barcodes setting.

If the number of recognized barcodes is equal to or more than the specified minimum, image enhancement is skipped. If the minimum number of barcodes is not reached, and if image enhancement is enabled, then barcode recognition is performed a second time. The recognition uses the configured type of image enhancement in an attempt to read the barcode.

For example, if the configured minimum barcode is 1 and the maximum is 10, first barcode recognition is performed without any image enhancement and attempt to read up to 10 barcodes. If at least one barcode is read, the action completes. If 0 barcodes are read, the action performs the specified image enhancement and attempts again to recognize the barcodes on the page. If 0 barcodes are read and no image enhancement is configured, the action completes without a second attempt at recognition.

Image Enhancement

Image enhancement might be necessary in order to help read damaged or poorly scanned barcodes. Depending on the problem with the barcode, different types of enhancements might make the barcode readable. The *barcode_p* actions offer a way to enhance the image before it reads the barcode. To enable this feature, set the variable *bp_enhance* to "1". This feature alters only the image for barcode recognition. The original image in the batch is left unchanged. The image enhancements are not cumulative, enable just the feature that is needed.

Depending on the type of image enhancement that is needed, you need to set the following extra variables:

- Smooth Zoom *bp_zoom* - This feature is automatically enabled when the variable *bp_enhance* is set to "1". To disable smooth zoom set the variable to "0". It is disabled automatically if one of the other image enhancement settings are enabled.

The following enhancement settings are disabled by default and can be enabled when the variable is set to "1":

- Blur *bp_blur* - Blurs the image, which can help jagged lines.
- Dilate *bp_dilate* - Expands the pixels, which make the lines thicker.
- Erode *bp_erode* - Removes pixels along line edges.
- Flip *bp_flip* - It is usually not necessary to flip the image for an upside-down barcode to recognize.
- Resize *bp_resize* - Enlarges the image, which can help small barcodes.

Note: Although image enhancement is turned on by setting the *bp_enhance* variable, enhancement is performed only if the number of recognized barcodes in the first recognition attempt is less than the minimum barcodes setting.

Barcode Orientation

Barcode orientation defaults to both horizontal and vertical. Specifying the orientation might help in detection of barcodes. If necessary, the orientation can be configured by setting the DCO variable *bp_or*.

0 = Horizontal and Vertical.

1 = Horizontal.

2 = Vertical.

3 = Horizontal, Vertical, and Diagonal.

Barcode Byte Mode

In rare cases, barcodes can contain null characters within the data stream. If null characters occur, the full length of the barcode can be truncated returning only the characters up to the first null character. To treat a barcode as byte data, set the DCO variable *BarcodeUseByte* to "1". This setting treats the stream as bytes and also convert any null characters to spaces so the entire string can then be utilized in other parts of the application.

When in byte mode, the default replacement character of a space can be set to a different replacement character by setting the DCO variable *BarcodeReplacementChar* to the desired replacement character.

Note: Unicode characters are not supported by the byte mode.

Example

```
GetBarcodeBP ()
```

Parent topic: [Barcode_P actions](#)

GetDataMatrixCodeBP

Recognizes Data Matrix codes

Member of namespace

Barcode_P

Syntax

```
bool GetDataMatrixCodeBP()
```

Parameters

None.

Returns

True if the action is called at the page level or field level. Otherwise, False.

In addition the calling object's value and variable *GetBarcode* is filled with the bar code value. This action also stores barcode information such as confidence, coordinates, code name, and size. If the object's barcode settings are set to read more than one barcode, and more than one barcode is found, barcodes are also stored in the variable *GetBarcodeX* where X is the index of the barcode found.

Level

Page or Field level only.

Details

Use this action if your page or field has Data Matrix codes. Any field must have a Position assigned when bar code reading is performed. For example the application would invoke the CreateFields action, and if the field has Position defined in the Document Hierarchy, it is ready for barcode reading. If anchors or fingerprint matching are used, ReadZones or other registration may be required to align the fields correctly.

Example:

```
GetDataMatrixCodeBP()
```

Parent topic: [Barcode_P actions](#)

IdentifyByBarcodesBP

Updates the current page type if a barcode match is found.

Syntax

```
bool IdentifyByBarcodesBP(string barcodePageMappings, string mappingsDelim, string keyValuePair, bool caseSensitive)
```

Parameters

barcodePageMappings

A delimited list of barcode value-page type mappings that are evaluated to determine whether matching barcodes exist on the current page. For more information, see the Details section.

If any of these mappings are regular expressions, set the batch variable *BarcodeRegex* to true. If this variable is not set to true, the regular expressions are not recognized as regular expressions.

mappingsDelim

(Optional) Delimiter separating mappings. Default: comma.

keyValueSep

(Optional) Delimiter that is used to separate the barcode value and the page type in an individual mapping. Default: equals sign (=).

caseSensitive

(Optional) True enforces case sensitivity when you evaluate the list of barcodes that are found on the page. Default: False.

Returns

True, if a match is found. Otherwise, False.

Level

Page level

Details

This action searches all of the barcodes that are found on the current page for a match from the provided mappings by using the *GetBarcodeList* variable. If a barcode value is found, this action updates the type of the current page by using the corresponding input mapping.

The GetAllBarcodesBP action is called if the *GetBarcodeList* variable does not exist. If the GetAllBarcodesBP action was called earlier with a string parameter, use an identical symbol for the mappingsDelim parameter.

If multiple barcode values are found, the page type that is assigned is the first corresponding value from the *GetBarcodeList* variable.

Example

```
IdentifyByBarcodesBP (Separator=Separator_PageAttach=Attachment_Page,,,)
IdentifyByBarcodesBP (Separator=Separator_Page%Attach=Attachment_Page,%,,,)
IdentifyByBarcodesBP (Separator|Separator_Page%Attach|Attachment_Page,%|, True)
```

Parent topic: [Barcode_P actions](#)

MatchBarcodeBP

Searches all the barcodes on the current page and checks if one of them matches the value that you entered as a parameter.

Member of namespace

Barcode_P

Syntax

```
bool MatchBarcodeBP (StrParam)
```

Parameters

The String value of the barcode.

Returns

True if the action is called at the page level or field level and one of the barcode values on the page matches the parameter value. The parameter value must not be empty. Otherwise, False.

In addition the calling object's value and variable *GetBarcode* is filled with the bar code value. This action also stores barcode information such as confidence, coordinates, code name, and size. If the object's barcode settings are set to read more than one barcode, and more than one barcode is found, barcodes are also stored in the variable *GetBarcodeX* where X is the index of the barcode found.

Level

Page level and field level.

Details

Searches all the barcodes on the current page and checks if one of the barcodes matches the parameter value. If a match occurs, the barcode's value is placed into a page level variable called *GetBarcode*. Refer to action *GetBarcode* for information regarding barcode configuration in Datacap Studio.

Example:

```
MatchBarcodeBP ("2008")
```

Parent topic: [Barcode_P actions](#)

Related reference:

[GetBarcodeBP](#)

MatchBarcodePrefixBP

Searches all the barcodes on the current page and checks if one of them matches the value that you entered as a parameter.

Member of namespace

Barcode_P

Syntax

```
bool MatchBarcodePrefixBP (StrParam)
```

Parameters

The String value of the barcode.

Returns

True if the action is called at the page level or field level and one of the barcode values on the page matches the parameter value. The parameter value must not be empty. Otherwise, False.

In addition the calling object's value and variable *GetBarcode* is filled with the bar code value. This action also stores barcode information such as confidence, coordinates, code name, and size. If the object's barcode settings are set to read more than one barcode, and more than one barcode is found, barcodes are also stored in the variable *GetBarcodeX* where X is the index of the barcode found.

Level

Page level and field level.

Details

Searches all the barcodes on the current page and checks if one of the barcodes starts with the parameter value. If a match occurs, the value of the barcode is placed into a page level variable called *GetBarCode*.

Refer to the *GetBarCode* action for information regarding barcode configuration in Datacap Studio.

Example:

```
MatchBarcodePrefixBP("ATM")
```

Parent topic: [Barcode_P actions](#)

Related reference:

[GetBarcodeBP](#)

ReadBarCodeBP

Tests to determine if the first barcode on the current page contains the value that is specified by the parameter.

Member of namespace

Barcode_P

Syntax

```
bool ReadBarCodeBP(StrParam)
```

Parameters

A single string value of the barcode.

Returns

True if the first barcode on the page has a value that matches the parameter. Otherwise, False.

Level

Page level only.

Details

Checks if the current page contains a barcode with the value specified by the parameter. This action uses the first barcode it encounters. One possible use of this action is to identify a document's Separator page. Refer to action *GetBarCode* for information regarding barcode configuration in Datacap Studio.

Example:

```
ReadBarCodeBP("Separator")  
SetPageType("Separator")
```

This example looks for a barcode with the value "Separator". If found, the second action, a DCO action, establishes the page as a Separator page.

Parent topic: [Barcode_P actions](#)

Related reference:

[GetBarcodeBP](#)

SetMinimumConfidenceBP

Sets the minimum confidence level that is required for barcodes that are read by the engine to be accepted.

Syntax

```
bool SetMinimumConfidenceBP(StrParam)
```

Parameters

An integer value that represents the minimum confidence level that is required for barcodes that are read by the engine to be accepted. Valid values are in the range of 1-10. The default value, when this engine is not used, is 7.

Returns

False if the confidence value cannot be set either because it is invalid or if an error occurs or if the action is called at a level other than page or field. Otherwise, True.

Level

Page or Field level only

Details

Sets the minimum confidence level that is required for barcodes that are read by the engine to be accepted.

Example

```
SetMinimumConfidenceBP(5)  
GetBarcodeBP()
```

Parent topic: [Barcode_P actions](#)

ClassifyLayout actions

Use the ClassifyLayout actions to identify page types based on page layout templates.

Important: These actions are technology preview code. Preview code is not intended for use in production; you use this code as is and without IBM support. If you use any preview code in your application, you might need to make updates to your application in later Datacap releases. Report any issues with the ClassifyLayout preview code on the [Datacap developerWorks forum](#).

A layout template consists of page layout features with an associated page type. Here are some of the layout features that are used:

- Title (required)

- Page number
- Average font size
- The presence and locations of tables and graphics

These templates are stored in the Fingerprint database.

- [Feedback](#)
Adds a layout template of the current page to the Fingerprint database.
- [Identify](#)
For page identification purposes, creates page variables whose values reflect the most closely matching layout template in the Fingerprint database for the current page.

Parent topic: [Global actions](#)

Feedback

Adds a layout template of the current page to the Fingerprint database.

Important: These actions are technology preview code. Preview code is not intended for use in production; you use this code as is and without IBM support. If you use any preview code in your application, you might need to make updates to your application in later Datacap releases. Report any issues with the ClassifyLayout preview code on the [Datacap developerWorks forum](#).

Member of namespace

ClassifyLayout

Syntax

```
bool Feedback ()
```

Returns

True if the action completes without errors. Otherwise, if the action is called for an object other than a page object or if an internal error occurs, the action returns False.

Level

Page level.

Details

Use this action in the following ways to train the application to identify page types from layout templates:

- To initially provide a base set of templates
- To incrementally add new templates to improve accuracy

After you train the application, you can use the Identify action to identify a new page by finding the most closely matching template.

Call this action in the following circumstances:

- After you confirm that the page type as specified by the Page type field is correct
- After you call the Identify action

Typically, you call this action at export time.

This action compares the values of the Page type field and the acidType variable (which is set by the Identify action). If the values are not the same, a layout template of the current page is added to the database.

Example

```
// Mandatory prerequisites
Recognize()
Identify()

// Give feedback
Feedback()
```

Parent topic: [ClassifyLayout actions](#)

Identify

For page identification purposes, creates page variables whose values reflect the most closely matching layout template in the Fingerprint database for the current page.

Important: These actions are technology preview code. Preview code is not intended for use in production; you use this code as is and without IBM support. If you use any preview code in your application, you might need to make updates to your application in later Datacap releases. Report any issues with the ClassifyLayout preview code on the [Datacap developerWorks forum](#).

Member of namespace

ClassifyLayout

Syntax

```
bool Identify()
```

Returns

True if the action completes without errors. Otherwise, if the action is called for an object other than a page object or if an internal error occurs, the action returns False.

Level

Page level.

Details

Use this action to search the previously learned layout templates to find the closest match to the current page layout. The layout must contain a title in order for matching to succeed. If a similar layout is found, the following page variables are set:

Page variable	Variable value
acidType	The associated page type of the most closely matching layout template.

Page variable	Variable value
acidPageNo	The page number of the most closely matching template. For example, if the most closely matching template was for the second page of a three page document, the acidPageNo value is "2/3". If the most closely matching template has no associated page number, the acidPageNo value is "0/0".
acidTitle	The page title, if any, of the most closely matching template.

Before you call this action, you must call an OCR action that produces an XML layout file for the current page. Two examples of such actions are the Recognize actions in the OCR_A and OCR_S libraries.

Example

```
// Mandatory prerequisite: An action that performs OCR for the page
// The OCR action must create an XML layout file (such as OCR_SR/recognize or
OCR_A/recognize)
Recognize()

// Identify page type
Identify()

// Optionally copy the acidType field value as the page type
rrSet("acidType", "Page type")
```

Parent topic: [ClassifyLayout actions](#)

CC actions

Use the CC actions to identify types of pages and other text with IBM® Content Classification technology.

In general, CC actions are oriented towards either the use of an IBM Content Classification knowledge base or the use of an IBM Content Classification decision plan. For more information, see [IBM Content Classification: Category and rule-based classification](#).

The CC actions can be classified as follows:

Table 1. CC methods by page identification method

	Category-based classification	Rule-based classification
Common settings	<ul style="list-style-type: none"> • SetListenerURLCC 	
Settings	<ul style="list-style-type: none"> • SetKnowledgeBaseCC • SetLanguageCC • SetProblemValueCC 	<ul style="list-style-type: none"> • SetDecisionPlanCC • SetDecisionPlanFieldsCC
Main methods	<ul style="list-style-type: none"> • ClassifyCC • ClassifyTextCC • UpdateKnowledgeBaseCC 	<ul style="list-style-type: none"> • RunDecisionPlanCC • RunDecisionPlanForBlocksCC • RunDecisionPlanForTextCC

The settings affect the behavior of the main methods.

- [ClassifyCC](#)
Identifies the page type for the current page by using category-based classification.
- [ClassifyTextCC](#)
Finds matching categories for the specified text by using category-based classification.

- [RunDecisionPlanCC](#)
For both page type identification and data import purposes, creates page variables by using rule-based classification. This classification is based on the page's text.
- [RunDecisionPlanForBlocksCC](#)
For both page type identification and data import purposes, creates page variables by using rule-based classification. The classification is based on a specified list of page text blocks that indicate the page text to use.
- [RunDecisionPlanForTextCC](#)
For both page type identification and data import purposes, creates page variables by using rule-based classification. The classification is based on the specified text.
- [SetDecisionPlanCC](#)
A setting for the name of the IBM Content Classification decision plan.
- [SetDecisionPlanFieldsCC](#)
A setting for a comma-separated field name list for those fields that are to be created on the current DCO object in accordance with rule-based classification.
- [SetKnowledgeBaseCC](#)
A setting for the name of the IBM Content Classification knowledge base to use.
- [SetLanguageCC](#)
A setting for the language to be used for finding matching categories in an IBM Content Classification knowledge base. Any categories in the knowledge base that do not belong to the specified language are excluded as possible matches.
- [SetListenerURLCC](#)
A setting for the URL of the IBM Content Classification listener.
- [SetProblemValueCC](#)
A setting for the minimum category match score, which is the minimum score for a category to be considered a match with a piece of text. Valid values are 0.0 – 1.0, inclusive. For example, you might set the minimum score to 0.9.
- [UpdateKnowledgeBaseCC](#)
Updates the CC Knowledge Base.
- [Category match variables](#)
Some CC actions set page variables on the calling DCO object to return data for the matching categories. What constitutes a matching category is determined by the minimum category match score (as set by the SetProblemValueCC action).

Parent topic: [Global actions](#)

ClassifyCC

Identifies the page type for the current page by using category-based classification.

Member of namespace

CC

Syntax

```
bool ClassifyCC ()
```

Returns

True if a category is matched. Otherwise, this action returns False.

Level

Page level.

Details

Use this action to set the following items:

Item	Description	Comments
Page type field	If matching categories are found, the value of the page type field is set to the ID of the matching category with the highest confidence score. Otherwise, if no matching categories are found, the page type is unchanged.	The page type is updated by this action only if the variable UpdateDCOType has a value other than "0". If you do not want the page type to be updated, set UpdateDCOType to "0" before you call this action.
Match variables	Some variables are set to contain category match data. For more information, see Category match variables .	

Example

```
// Mandatory prerequisite: An action that performs OCR for the page
Recognize()

// Mandatory settings
SetListenerURLCC("http://localhost:18087")
SetKnowledgeBaseCC("Mortgage")
SetLanguageCC("English")

// Optional settings
SetProblemValueCC(0.9)

// Identify the page type
ClassifyCC()
```

Parent topic: [CC actions](#)

ClassifyTextCC

Finds matching categories for the specified text by using category-based classification.

Member of namespace

CC

Syntax

```
bool ClassifyTextCC(string TextToClassify)
```

Parameters

TextToClassify

The text to be categorized.

Smart parameters are supported.

Returns

True if the action completes successfully. Otherwise, this action returns False.

Level

All levels.

Details

Use this action to set the following items:

Item	Description
Match variables	Some variables are set to contain category match data. For more information, see Category match variables .

Example

```
// Mandatory prerequisite: An action that performs OCR for the page
// The OCR action must create an XML layout file (such as OCR_SR/recognize or
OCR_A/recognize)
Recognize()

// Mandatory settings
SetListenerURLCC("http://localhost:18087")
SetKnowledgeBaseCC("Mortgage")
SetLanguageCC("English")

// Optional settings
SetProblemValueCC(0.9)

// Find matching categories
ClassifyTextCC("Place text to classify here")
```

Parent topic: [CC actions](#)

RunDecisionPlanCC

For both page type identification and data import purposes, creates page variables by using rule-based classification. This classification is based on the page's text.

Member of namespace

CC

Syntax

```
bool RunDecisionPlanCC()
```

Returns

True if the action completes successfully. Otherwise, this action returns False.

Level

Page level.

Details

Use this action to set the following items:

Item	Description
Page variables	The page's text is run against the decision plan. (The decision plan is specified by the SetDecisionPlanCC action). Page variables are created in accordance with the SetDecisionPlanFieldsCC setting and the rules of the decision plan. For more information, see IBM Content Classification: Category and rule-based classification .
Match variables	Some variables are set to contain category match data. For more information, see Category match variables .

Example

```
// Mandatory prerequisite: An action that performs OCR for the page
Recognize ()

// Mandatory settings
SetListenerURLCC ("http://localhost:18087")
SetDecisionPlanCC ("Mortgage")
SetDecisionPlansFieldsCC ("EmployeeID.Date")

// Run the decision plan
RunDecisionPlanCC ()
```

Parent topic: [CC actions](#)

RunDecisionPlanForBlocksCC

For both page type identification and data import purposes, creates page variables by using rule-based classification. The classification is based on a specified list of page text blocks that indicate the page text to use.

Member of namespace

CC

Syntax

```
bool RunDecisionPlanForBlocksCC (string blockTypes)
```

Parameters

blockTypes

A comma-separated list of one or more of the following block types:

- Block
- Table
- Header
- Footer
- Title
- H1

- H2
- H3
- Barcode

For general information about block types, see [DocumentAnalytics actions](#).

Important:

- Block types must be defined as input fields in the decision plan. For example, to use the Barcode block type, a Barcode input field must be defined in the decision plan. For information about adding input fields to a decision plan and mapping XML nodes to them, see [Configuring rule-based classification by using a decision plan](#).
- Depending on the version of IBM® Content Classification that you installed, you might need to make some XML configuration file changes. These changes are required to ensure that the IBM Content Classification server properly parses text blocks. For more information, see [Configuring content classification for XML layout block parsing](#).

Smart parameters are supported.

Returns

True if the action completes successfully. Otherwise, this action returns False.

Level

Page level.

Details

Use this action to set the following items:

Item	Description
Page variables	Based on the specified block types and the page's XML layout file, a portion of the page's text is run against the decision plan. (The decision plan is specified by the SetDecisionPlanCC action). For example, suppose that the blockTypes parameter specifies the following block type only: <i>Table</i> . In this case, the page's text that is run against the decision plan consists of those page text blocks that belong to the block type <i>Table</i> . Page text in other text blocks is ignored. Page variables are created in accordance with the SetDecisionPlanFieldsCC setting and the rules of the decision plan. For more information, see IBM Content Classification: Category and rule-based classification .
Match variables	Some variables are set to contain category match data. For more information, see Category match variables .

Example

```
// Mandatory prerequisite: An action that performs OCR for the page
// The OCR action must create an XML layout file (such as OCR_SR/recognize or
OCR_A/recognize)
Recognize()

// Mandatory settings
SetListenerURLCC("http://localhost:18087")
SetKnowledgeBaseCC("Mortgage")
SetDecisionPlanCC("Mortgage")
```

```
SetDecisionPlansFieldsCC("Title,Loan,Loan Type")

// Run decision plan
RunDecisionPlanForBlocksCC("Table,Block")
```

Parent topic: [CC actions](#)

RunDecisionPlanForTextCC

For both page type identification and data import purposes, creates page variables by using rule-based classification. The classification is based on the specified text.

Member of namespace

CC

Syntax

```
bool RunDecisionPlanForTextCC(string TextToClassify)
```

Parameters

TextToClassify

The text to run against the decision plan.

Smart parameters are supported.

Returns

True if the action completes successfully. Otherwise, this action returns False.

Level

All levels.

Details

Use this action to set the following items:

Item	Description
Page variables	The specified text is run against the decision plan. (The decision plan is specified by the SetDecisionPlanCC action). Page variables are created in accordance with the SetDecisionPlanFieldsCC setting and the rules of the decision plan.
Match variables	Some variables are set to contain category match data. For more information, see Category match variables .

Example

```
// Mandatory prerequisite: An action that performs OCR for the page
Recognize()

// Mandatory settings
SetListenerURLCC("http://localhost:18087")
SetKnowledgeBaseCC("Mortgage")
```

```
SetDecisionPlanCC("Mortgage")
SetDecisionPlansFieldsCC("Title,Loan,Loan Type")

// Run the decision plan
RunDecisionPlanForTextCC(@F)
```

Parent topic: [CC actions](#)

SetDecisionPlanCC

A setting for the name of the IBM® Content Classification decision plan.

Smart parameters are supported.

Member of namespace

CC

Syntax

```
bool SetDecisionPlanCC(string DecisionPlanName)
```

Returns

False if the parameter value is empty. Otherwise, this action returns True.

Level

All levels.

Details

Example

```
SetDecisionPlanCC("Mortgage")
```

Parent topic: [CC actions](#)

SetDecisionPlanFieldsCC

A setting for a comma-separated field name list for those fields that are to be created on the current DCO object in accordance with rule-based classification.

Member of namespace

CC

Syntax

```
bool SetDecisionPlanFieldsCC(string DPFields)
```

Returns

False if the parameter value is empty. Otherwise, this action returns True.

Level

All levels.

Details

Example

```
SetDecisionPlanFieldsCC("Title,Loan,Loan Type")
```

Parent topic: [CC actions](#)

SetKnowledgeBaseCC

A setting for the name of the IBM® Content Classification knowledge base to use.

Smart parameters are supported.

Member of namespace

CC

Syntax

```
bool SetKnowledgeBaseCC(string KnowledgeBaseName)
```

Returns

False if the parameter value is empty. Otherwise, this action returns True.

Level

All levels.

Details

Example

```
// Set the name of the knowledge base  
SetKnowledgeBaseCC("Mortgage")
```

Parent topic: [CC actions](#)

SetLanguageCC

A setting for the language to be used for finding matching categories in an IBM® Content Classification knowledge base. Any categories in the knowledge base that do not belong to the specified language are excluded as possible matches.

Member of namespace

CC

Syntax

```
bool SetLanguageCC(string LanguageName)
```

Returns

Always True.

Level

All levels.

Details

Example

```
SetLanguageCC("English")
```

Parent topic: [CC actions](#)

SetListenerURLCC

A setting for the URL of the IBM® Content Classification listener.

Smart parameters are supported.

Member of namespace

CC

Syntax

```
bool SetListenerURLCC(string URL)
```

Returns

True if the URL is successfully set. Otherwise, this action returns False. This action does not test that the listener exists and is running.

Level

All.

Details

Example

```
SetListenerURLCC("http://localhost18087")
```

Parent topic: [CC actions](#)

SetProblemValueCC

A setting for the minimum category match score, which is the minimum score for a category to be considered a match with a piece of text. Valid values are 0.0 – 1.0, inclusive. For example, you might set the minimum score to 0.9.

Tip: For the passed parameter value, use the appropriate decimal character for your system locale as defined for your application in the Datacap Application Manager. For example, if the decimal character is a comma, you would specify the minimum score as 0,9 instead of 0.9.

Member of namespace

CC

Syntax

```
bool SetProblemValueCC(double MinScore)
```

Returns

True if the parameter value is valid. Otherwise, this action returns False.

Level

All levels.

Details

Example

```
SetProblemValueCC(0.9)
```

Parent topic: [CC actions](#)

UpdateKnowledgeBaseCC

Updates the CC Knowledge Base.

Member of namespace

CC

Syntax

```
bool UpdateKnowledgeBaseCC()
```

Parameters

None.

Returns

True if the IBM® Content Classification Knowledge Base is successfully updated. Otherwise, False.

Level

Page level.

Details

Provides classification feedback to the IBM Content Classification server.

Important: This method is to be used to provide feedback to a previously trained knowledge base, and not to be used as the primary method for training a knowledge base from scratch.

Although training via feedback is not recommended to build the knowledge base, it is recommended that it is used to submit feedback on classification data returned on a "trained" category. For instance, let us use a knowledge base that contains 5 categories, 4 of which have been trained. When ClassifyCC runs, it will either return a match or no match. When a match is found, it is recommended to provide feedback (using UpdateKnowledgeBaseCC) to the knowledge base to assert that the classification was correct.

When there is no match (no category returned) or there is a mismatch (wrong category returned), the recommended process is to export a copy of the document to a folder and train the knowledge base with these new documents via the IBM Content Classification Workbench. After the new documents have been added, the knowledge based should be exported, and documents processed in Datacap to provide feedback. The training should be done in a non Production system, then moved to Production after testing.

Although it is recommended that this procedure is used to submit feedback on trained categories, it should be done for a period of time, and not indefinitely, since this can cause the knowledge base to grow very large.

In conclusion, this action should be used only for documents where the classification result was correct, and should only be used during a period of time until the training on the knowledge base is complete.

Example

```
// Mandatory settings
SetListenerURLCC("http://localhost:18087")
SetKnowledgeBaseCC("Mortgage")
ClassifyCC()

// Update the knowledge base
UpdateKnowledgeBaseCC()
```

Parent topic: [CC actions](#)

Category match variables

Some CC actions set page variables on the calling DCO object to return data for the matching categories. What constitutes a matching category is determined by the minimum category match score (as set by the SetProblemValueCC action).

The following table lists these variables:

MatchingCategoriesCount	The number of matching categories
MatchingCategoryX	A list of the matching categories
MatchingCategoryConfX	A list of the matching category scores

Parent topic: [CC actions](#)

Cco2cco actions

Use the Cco2cco actions to sort and filter the words and lines in a fingerprint CCO file.

This normalization is only required after full page recognition by an OCR or ICR action that does not automatically normalize the CCO. The OCR/S, OCR/A, and ICR/C actions normalize automatically.

- [NormalizeCCO](#)
The NormalizeCCO action sorts and filters the words and lines in a Fingerprint file (.cco) that are created by a Recognition engine, for use by navigation and pattern match actions. This action is only required after full page recognition by an OCR or ICR action that does not automatically normalize the CCO.
- [SetMaxCharacterHeightAVG](#)
Sets the maximum height of characters, by percentage over the average, that is allowed by the Cco2cco and NormalizeCCO actions.
- [SetMaxCharacterHeightTMM](#)
Sets the maximum height of characters, by absolute value, that is allowed by the Cco2cco and NormalizeCCO actions.

Parent topic: [Global actions](#)

NormalizeCCO

The NormalizeCCO action sorts and filters the words and lines in a Fingerprint file (.cco) that are created by a Recognition engine, for use by navigation and pattern match actions. This action is only required after full page recognition by an OCR or ICR action that does not automatically normalize the CCO.

Member of namespace

Cco2cco

Syntax

```
bool NormalizeCCO ()
```

Returns

Always True.

Level

Page.

Details

This action sorts the words and lines in a Fingerprint file (.cco) created by a recognition engine, for use by navigation and pattern match actions. The action is called by full-page recognition actions for ICR/C, OCR/S, and OCR/A. This action must always be called before the Locate actions or the pat_RecogMatch_ID action are used to find recognized text on a page.

In this context, the fingerprint is calculated for a particular image in a batch, not in the Fingerprint database that contains fingerprints for various page types and layout variations that are defined for a particular application.

There are two types of Fingerprint files. One type is based on the image geometry. The second type is based on recognized text. The AnalyzeImage action creates a geometric fingerprint that contains lines and words that are based only on the black pixels in the image. Full-page recognition actions, such as RecognizePageOCR_S, RecognizePageICR_C, RecognizePageOCR_A, create a fingerprint that is based on the results of recognition; that is, both geometry and text of the recognized characters, words and lines.

In Recognition-based fingerprints, the order of lines and words might appear to be arbitrary, especially if the page contains images, tables, stamps, or blocks of text with varying font sizes. This can cause unpredictable results from Locate actions that navigate geometrically. The word-matching and phrase-matching action pat_RecogMatch_ID also requires well-ordered text to work reliably.

The NormalizeCCO action reorders the words of text in a Recognition-based fingerprint into lines and words in standard reading order, from top to bottom and left to right.

Important: NormalizeCCO discards any "words" or blocks that contain characters taller than 1/4 inch, or the height set by SetMaxCharacterHeightTMM().

If the AnalyzeImage action is called before full-page recognition, the recognized text is placed into the geometry that is created by AnalyzeImage. This hybrid Fingerprint file is not always suitable for cco2cco. To force creation of a pure recognition-based fingerprint, call SetFingerprintRecogPriority(True) before full-page recognition. This guarantees that any existing geometric fingerprint is ignored, and it applies to OCR_S and ICR_C only.

The full page recognition actions from the ICR_C, OCR_A, and OCR_S libraries call NormalizeCCO() automatically unless the action CCONormalization_OFF (from the Recog_Shared library) is called before recognition. The full page recognition from the OCR_SR library, however, requires that NormalizeCCO() to be called manually post recognition.

Example:

```
SetFingerprintRecogPriority(True)
RecognizePageOCR_S()
NormalizeCCO()
pat_RecogMatch_ID()
```

Parent topic: [Cco2cco actions](#)

SetMaxCharacterHeightAVG

Sets the maximum height of characters, by percentage over the average, that is allowed by the Cco2cco and NormalizeCCO actions.

Member of namespace

Cco2cco

Syntax

```
bool SetMaxCharacterHeightAVG(strParam)
```

Parameters

One or two digit Integer value sets the percent maximum height of characters over the average to permit in the CCO. Off by default. Parameter of zero (0) or less turns off this functionality.

Returns

Always True.

Level

Page.

Details

Example:

```
SetMaxCharacterHeightAVG (15)
NormalizeCCO ()
```

Parent topic: [Cco2cco actions](#)

SetMaxCharacterHeightTMM

Sets the maximum height of characters, by absolute value, that is allowed by the Cco2cco and NormalizeCCO actions.

Member of namespace

Cco2cco

Syntax

```
bool SetMaxCharacterHeightTMM(strParam)
```

Parameters

Integer: maximum height of characters to permit in the CCO, in 1/10 mm units. Default is 64, or approx 1/4 inch (50 pixels at 200 DPI).

Returns

Always True.

Level

Page.

Details

Example:

```
SetMaxCharacterHeightTMM (75)
NormalizeCCO ()
```

Parent topic: [Cco2cco actions](#)

CheckProcessing actions

Use the Check Processing actions to extract data from checks originating from Argentina, Brazil, Canada, France, India, UK and US.

Supported image formats

This library supports single page TIF and JPEG images with a resolution of 200 DPI to 300 DPI.

US checks

The image must be black and white (1 bpp) or grayscale (8 bpp). Color is not supported for US checks. If you have colored US checks, these checks must be converted prior to calling recognition.

Brazilian checks

The checks can be black and white (1 bpp), 8 bit grayscale (8bpp) or 24 bit color (24 bpp) images. The Brazilian engine uses the color information when performing recognition, but color is not required.

Other international countries

For other countries, the checks can be black and white (1 bpp), 8 bit grayscale (8bpp) or 24 bit color (24 bpp) images. However, the color information is not used. The engine automatically converts the image to black and white prior to performing recognition. If you provide images with 24 bit color or 8 bit gray scale to the engines, these images are automatically binarized in memory before the recognition, leaving the original image unchanged. If desired, you can use actions to binarize images before processing, so you have a black and white image of the check saved within the batch.

Check Borders

The check image must be the size of the check itself. You must crop any borders or white space around the check image making the image the same size as the physical check or recognition might not be correct. You can use the image enhancement ruleset to deskew and remove the check border.

The following shows a specific example ruleset and functions configured to process checks originating from the US. The page is first converted to black and white, and then run through the check processing action.

```
Ruleset Detect Checks
Function Process Check
  OCR_AConvertImageToBW("tio")
  ProcessCheck("USA", "false", "None")
```

- [ProcessCheck](#)
Performs data extraction from checks originating from various regions.
- [CheckProcessingBrazil](#)
Performs data extraction from checks originating from Brazil.
- [CheckProcessingCanada](#)
Performs data extraction from checks originating from Canada.
- [CheckProcessingFrance](#)
Performs data extraction from checks originating from France.
- [CheckProcessingIndia](#)
Performs data extraction from checks originating from India.
- [CheckProcessingUK](#)
Performs data extraction from checks originating from the United Kingdom.
- [CheckProcessingUs](#)
Performs data extraction from checks originating from the United States.

Parent topic: [Global actions](#)

ProcessCheck

Performs data extraction from checks originating from various regions.

Syntax

```
bool ProcessCheck (string country, string createFields, string  
checkBackSideImagePath)
```

Parameters

country:

The three-letter ISO code that identifies the country for the check to recognize. The following countries are supported:

- Argentina
- Brazil
- Canada
- France
- India
- UK
- US

createFields:

When set to true, runtime DCO fields are created with the extracted data.

checkBackSideImagePath:

The path to the backside check image. If you do not provide image path, then the next subsequent page is used at the back image. To capture only fields on the front of the check, use the value "None". This parameter is used only when the country is India. The back side image parameter is ignored for all other countries.

Smart parameters are supported.

Level

Page

Returns

True, if the current image is a check. Otherwise, False.

Details

Detects check images and stores the extracted data into variables at the page level. If a field is not read successfully by the recognition engine, it is not included in the results. If a country supports recognition of the Payee, then a vocabulary is required. Payee Vocabularies are created as a comma separated text file in the following format:

"word or phrase 1",weight

"word or phrase 2",weight

"word or phrase 3",weight

"word or phrase 4",weight

The weight is an integer value from 1 to 100, inclusive. The words that are expected should be given a higher weight and words expected less often should be given a lower weight. The DCO variable `payeeVocab` must be set on the page level object with the full path to the vocabulary file.

The following example shows how to specify the vocabulary file using smart parameters. The vocabulary file can exist anywhere and this example shows how to access a file that has been placed in the application's DCO directory.

```
rrSet("@PROCESSDIR+\MyVocabulary.txt", "@X.payeeVocab")
```

When a vocabulary is provided, an optional DCO page variable, `payeeVocabCoverage`, can be set with an integer value from 1 to 100. This variable defines the approximate percentage of input payee names that match entries in a specified vocabulary for payee line recognition. If `payeeVocabCoverage` is not specified, then default is set to 50 percent.

Recognized check fields for each country

Argentina (ARG)

Detects check images and stores the following extracted data into variables at the page level:

- Amount
- MICR

The following runtime DCO fields are created:

- Amount
- AmountFull
- CAR
- LAR
- MICR
- MICRAccountNumber
- MICRCheckNumber
- MICRRoutingNumber
- MICRData0
- MICRData1
- MICRData2

Brazil (BRA)

Detects check images and stores the following extracted data into variables at the page level:

- Amount
- Date
- AmountVerification
- CAR
- LAR
- LARCARMismatch
- MICR
- PayorId
- Place
- TopLine
- PresenceFront

The following runtime DCO fields are created:

- Amount
- AmountFull
- CAR
- LAR
- Date
- YearPresence
- MonthPresence
- DayPresence
- Interval
- MICR
- MICRData0
- MICRData1
- MICRData2
- AmountVerification
- CARAV
- LARAV
- PresenceFront
- TopLine
- C1
- C2
- C3
- PayorID
- Place

Canada (CAN)

Detects check images and stores the following extracted data into variables at the page level:

- Amount
- Date
- MICR

The following runtime DCO fields are created:

- Amount
- AmountFull
- CAR
- LAR
- Date
- YearPresence
- MonthPresence
- DayPresence
- Interval
- MICR
- MICRData0
- MICRData1
- MICRData2

France (FRA)

Detects check images and stores the following extracted data into variables at the page level:

- Amount
- Date

- AmountVerification
- CAR
- LAR
- LARCARMismatch
- MICR
- CheckSum
- PresenceFront

The following runtime DCO fields are created:

- Amount
- AmountFull
- CAR
- LAR
- Date
- YearPresence
- MonthPresence
- DayPresence
- Interval
- MICR
- MICRData0
- MICRData1
- MICRData2
- AmountVerification
- CARAV
- LARAV
- PresenceFront
- CheckSum

India (IND)

Detects check images and stores the following extracted data into variables at the page level:

- AccountNoFront
- AccountNoBack
- Amount
- Date
- AmountVerification
- CAR
- LAR
- LARCARMismatch
- CtsCompliance
- CurrencySymbol
- DdMmYyyyCompliance
- VerticalTextPresence
- DateBoxes
- MICR

Note: For India, the front and back check information can be recognized. The front image is obtained from the image belonging to the current page object. The back image is specified as the second parameter to this action. If a back image is not specified, then the image belonging to the next consecutive page is used. To capture the fields on the front of the check only, use the parameter "None" for the back check image.

UK (GBR)

Detects check images and stores the following extracted data into variables at the page level:

- Amount
- CAR
- LAR

The following runtime DCO fields are created:

- Amount
- AmountFull
- CAR
- LAR

US (USA)

Detects check images and stores the following extracted data into variables at the page level:

- Amount
- Date
- MICR
- CAR
- LAR
- DocumentType

The following runtime DCO fields are created:

- Amount
- Date
- Year
- Month
- Day
- MICR
- MICRAccountNumber
- MICRCheckNumber
- MICRRoutingNumber
- MICRAmount
- MICRData0
- MICRData1
- MICRData2
- CheckNumber
- PayeeLine
- CARLARMismatch

Note: The DCO fields may not be created or will be blank, if the text is not recognized or if the information is not on the check. For example, typically checks do not contain the amount in the MICR field. For any country that supports CAR/LAR, the engine attempts to recognize the LAR (Legal Amount) only if the CAR (Courtesy Amount) is not recognized or is recognized with low confidence. The DCO variable `courtesyAcceptLevel` can be set for US check processing to indicate the minimum confidence needed before attempting LAR recognition. The `courtesyAcceptLevel` can be set within the confidence range of 0 to 100. The Amount contains the recognized amount with a decimal separator. The AmountFull contains the amount without a decimal separator.

The MICRData fields contain the separate logical components of the recognized MICR, such as the Account Number, separated out into separate variables. If the MICR is not successfully recognized, then these variables may not exist.

MICR font CMC-7 and E-13B are supported. The CMC-7 font is used in Argentina, Brazil, France, and the E-13B font is used in Canada, India, UK, and US.

This example is a page level rule that performs check recognition on the current page for a French check.

```
ProcessCheck("FRA", "true", "None")
```

Parent topic: [CheckProcessing actions](#)

CheckProcessingBrazil

Performs data extraction from checks originating from Brazil.

Important: This action is deprecated and scheduled to be removed in a future release. Use the ProcessCheck action in the CheckProcessing library. For more information see [ProcessCheck](#).

This library supports single page, black and white or 8 bit gray-scale TIF and images, or color JPEG images with a resolution of 200-300 DPI. If the images are colored or 8 bit gray scale, these images are automatically binarized in memory before the recognition, leaving the original image untouched. You can use actions to binarize images before processing so you have a saved black and white image. You can use a separate action to convert to black and white before performing check recognition.

CAUTION:

The binarization of the separate action may not be identical to the method used during recognition.

The check image must be the size of the check itself. You must crop any borders or white space around the check image making the actual image the same size, as the check recognition might not be correct.

The following shows a specific example ruleset and functions configured to process checks originating from Brazil. The page is first converted to black and white, and then run through the check processing action.

```
Ruleset Detect Checks
Function Process Check
    OCR_AConvertImageToBW("tio")
    ProcessCheckBrazil(false)
```

- [ProcessCheckBrazil](#)

Parent topic: [CheckProcessing actions](#)

ProcessCheckBrazil

Important: This action is deprecated and scheduled to be removed in a future release. Use the ProcessCheck action in the CheckProcessing library. For more information, see [ProcessCheck](#).

Member of namespace

CheckProcessing

Syntax

```
bool ProcessCheckBrazil (bool createFields)
```

Parameters

createFields:

When set to true, runtime DCO fields are created with the extracted data.

Details

Detects check images and stores the following extracted data into variables at the page level:

- Amount
- Date
- AmountVerification
- CAR
- LAR
- LARCARMismatch
- MICR
- PayorId
- Place
- TopLine
- PresenceFront

Level

Page.

Returns

True, if the current image is a check. Otherwise, False.

```
OCR_AConvertImage2BW("tio")
ProcessCheckBrazil(true)
```

Parent topic: [CheckProcessingBrazil](#)

CheckProcessingCanada

Performs data extraction from checks originating from Canada.

Important: This action is deprecated and scheduled to be removed in a future release. Use the `ProcessCheck` action in the `CheckProcessing` library. For more information see [ProcessCheck](#).

This library supports single page, black and white or 8 bit gray-scale TIF and images, or color JPEG images with a resolution of 200-300 DPI. If the images are colored or 8 bit gray scale, these images are automatically binarized in memory before the recognition, leaving the original image untouched. You can use actions to binarize images before processing so you have a saved black and white image. You can use a separate action to convert to black and white before performing check recognition.

CAUTION:

The binarization of the separate action may not be identical to the method used during recognition.

The check image must be the size of the check itself. You must crop any borders or white space around the check image making the actual image the same size, as the check recognition might not be correct.

The following shows a specific example ruleset and functions configured to process checks originating from Canada. The page is first converted to black and white, and then run through the check processing action.

```
Ruleset Detect Checks
Function Process Check
    OCR_AConvertImageToBW("tio")
    ProcessCheckCanada(false)
```

- [ProcessCheckCanada](#)

Parent topic: [CheckProcessing actions](#)

ProcessCheckCanada

Important: This action is deprecated and scheduled to be removed in a future release. Use the ProcessCheck action in the CheckProcessing library. For more information, see [ProcessCheck](#).

Member of namespace

CheckProcessing

Syntax

```
bool ProcessCheckCanada (bool createFields)
```

Parameters

createFields:

When set to true, runtime DCO fields are created with the extracted data.

Details

Detects check images and stores the following extracted data into variables at the page level:

- Amount
- Date
- MICR

Level

Page.

Returns

True, if the current image is a check. Otherwise, False.

```
OCR_AConvertImage2BW("tio")  
ProcessCheckCanada(true)
```

Parent topic: [CheckProcessingCanada](#)

CheckProcessingFrance

Performs data extraction from checks originating from France.

Important: This action is deprecated and scheduled to be removed in a future release. Use the ProcessCheck action in the CheckProcessing library. For more information see [ProcessCheck](#).

This library supports single page, black and white or 8 bit gray-scale TIF and images, or color JPEG images with a resolution of 200-300 DPI. If the images are colored or 8 bit gray scale, these images are automatically binarized in memory before the recognition, leaving the original image untouched. You can use actions to binarize images before processing so you have a saved black and white image. You can use a separate action to convert to black and white before performing check recognition.

CAUTION:

The binarization of the separate action may not be identical to the method used during recognition.

The check image must be the size of the check itself. You must crop any borders or white space around the check image making the actual image the same size, as the check recognition might not be correct.

The following shows a specific example ruleset and functions configured to process checks originating from France. The page is first converted to black and white, and then run through the check processing action.

```
Ruleset Detect Checks
Function Process Check
  OCR_AConvertImageToBW("tio")
  ProcessCheckFrance(false)
```

- [ProcessCheckFrance](#)

Parent topic: [CheckProcessing actions](#)

ProcessCheckFrance

Important: This action is deprecated and scheduled to be removed in a future release. Use the [ProcessCheck](#) action in the [CheckProcessing](#) library. For more information, see [ProcessCheck](#).

Member of namespace

CheckProcessing

Syntax

```
bool ProcessCheckFrance (bool createFields)
```

Parameters

createFields:

When set to true, runtime DCO fields are created with the extracted data.

Details

Detects check images and stores the following extracted data into variables at the page level:

- Amount
- Date
- AmountVerification
- CAR
- LAR
- LARCARMismatch
- MICR
- CheckSum
- PresenceFront

Level

Page.

Returns

True, if the current image is a check. Otherwise, False.

```
OCR_AConvertImage2BW("tio")  
ProcessCheckFrance(true)
```

Parent topic: [CheckProcessingFrance](#)

CheckProcessingIndia

Performs data extraction from checks originating from India.

Important: This action is deprecated and scheduled to be removed in a future release. Use the ProcessCheck action in the CheckProcessing library. For more information see [ProcessCheck](#).

This library supports single page, black and white or 8 bit gray-scale TIF and images, or color JPEG images with a resolution of 200-300 DPI. If the images are colored or 8 bit gray scale, these images are automatically binarized in memory before the recognition, leaving the original image untouched. You can use actions to binarize images before processing so you have a saved black and white image. You can use a separate action to convert to black and white before performing check recognition.

CAUTION:

The binarization of the separate action may not be identical to the method used during recognition.

The check image must be the size of the check itself. You must crop any borders or white space around the check image making the actual image the same size, as the check recognition might not be correct.

The following shows a specific example ruleset and functions configured to process checks originating from India. The page is first converted to black and white, and then run through the check processing action.

```
Ruleset Detect Checks  
  Function Process Check  
    OCR_AConvertImageToBW("tio")  
    ProcessCheckIndia(false)
```

- [ProcessCheckIndia](#)

Parent topic: [CheckProcessing actions](#)

ProcessCheckIndia

Important: This action is deprecated and scheduled to be removed in a future release. Use the ProcessCheck action in the CheckProcessing library. For more information, see [ProcessCheck](#).

Member of namespace

CheckProcessing

Syntax

```
bool ProcessCheckIndia (bool createFields, string checkBackSideImagePath)
```

Parameters

createFields

When set to true, runtime DCO fields are created with the extracted data.

checkBackSideImagePath (required)

Full path to the image for the back side of the check. Smart parameter supported.

Details

Detects check images and stores the following extracted data into variables at the page level:

- AccountNoFront
- AccountNoBack
- Amount
- Date
- AmountVerification
- CAR
- LAR
- LARCARMismatch
- CtsCompliance
- CurrencySymbol
- DdMmYyyyCompliance
- VerticalTextPresence
- DateBoxes
- MICR

Note: This action requires a front and back check image. The front image is obtained from the image that belongs to the current page object. The back image is specified as the second parameter to this action. If a back image is not specified, then the image that belongs to the next consecutive page is used.

Level

Page.

Returns

True, if the current image is a check and if the back image is also present. Otherwise, False.

```
OCR_AConvertImage2BW("tio")
ProcessCheckIndia(true, "c:\datacap\checks\batches\20150102.001\tm0000001_back.tif")
```

Parent topic: [CheckProcessingIndia](#)

CheckProcessingUK

Performs data extraction from checks originating from the United Kingdom.

Important: This action is deprecated and scheduled to be removed in a future release. Use the `ProcessCheck` action in the `CheckProcessing` library. For more information see [ProcessCheck](#).

This library supports single page, black and white or 8 bit gray-scale TIF and images, or color JPEG images with a resolution of 200-300 DPI. If the images are colored or 8 bit gray scale, these images are automatically binarized in memory before the recognition, leaving the original image untouched. You can use actions to binarize images before processing so you have a saved black and white image. You can use a separate action to convert to black and white before performing check recognition.

CAUTION:

The binarization of the separate action may not be identical to the method used during recognition.

The check image must be the size of the check itself. You must crop any borders or white space around the check image making the actual image the same size, as the check recognition might not be correct.

The following shows a specific example ruleset and functions configured to process checks originating from the United Kingdom. The page is first converted to black and white, and then run through the check processing action.

```
Ruleset Detect Checks
Function Process Check
  OCR_AConvertImageToBW("tio")
  ProcessCheckUK(false)
```

- [ProcessCheckUK](#)

Parent topic: [CheckProcessing actions](#)

ProcessCheckUk

Important: This action is deprecated and scheduled to be removed in a future release. Use the [ProcessCheck](#) action in the [CheckProcessing](#) library. For more information, see [ProcessCheck](#).

Member of namespace

CheckProcessing

Syntax

```
bool ProcessCheckUk (bool createFields)
```

Parameters

createFields:

When set to true, runtime DCO fields are created with the extracted data.

Details

Detects check images and stores the following extracted data into variables at the page level:

- AccountNoFront
- AccountNoBack
- Amount
- Date
- AmountVerification
- CAR
- LAR
- LARCARMismatch
- CheckSum
- PresenceFront

Level

Page.

Returns

True, if the current image is a check. Otherwise, False.

```
OCR_AConvertImage2BW("tio")
ProcessCheckUk(true)
```

Parent topic: [CheckProcessingUK](#)

CheckProcessingUs

Performs data extraction from checks originating from the United States.

Important: This action is deprecated and scheduled to be removed in a future release. Use the `ProcessCheck` action in the `CheckProcessing` library. For more information see [ProcessCheck](#).

This library supports single-page, black-and-white or 8-bit gray-scale TIF images, or color JPEG images with a resolution of 200-300 DPI. If the images are colored or 8 bit gray scale, these images are automatically binarized in memory before the recognition, leaving the original image untouched. You can use actions to binarize images before processing so you have a saved black and white image. You can use a separate action to convert to black and white before performing check recognition.

CAUTION:

The binarization of the separate action may not be identical to the method used during recognition.

The check image must be the size of the check itself. You must crop any borders or white space around the check image making the actual image the same size, as the check recognition might not be correct.

The following shows a specific example ruleset and functions configured to process checks originating from the United States. The page is first converted to black and white, and then run through the check processing action.

```
Ruleset Detect Checks
Function Process Check
    OCR_AConvertImageToBW("tio")
    ProcessCheckUs(false)
```

- [ProcessCheckUs](#)
Detects check images and stores extracted data into variables at the page level.

Parent topic: [CheckProcessing actions](#)

ProcessCheckUs

Detects check images and stores extracted data into variables at the page level.

Important: This action is deprecated and scheduled to be removed in a future release. Use the `ProcessCheck` action in the `CheckProcessing` library. For more information, see [ProcessCheck](#).

Member of namespace

`CheckProcessing`

Syntax

```
bool ProcessCheckUs (bool createFields)
```

Parameters

`createFields:`

When set to true, runtime DCO fields are created with the extracted data.

`engine`

An integer value that determines which engine to use. The default value is 0, which uses the CheckPlus Engine. This engine has an 82% read rate, and 1% error rate. It extracts the check type, amount, MICR, date, and check number.

Details

Detects check images and stores the following extracted data into variables at the page level:

- Amount
- Date
- MICR (when read successfully)
- CAR (when read successfully)
- LAR (when read successfully)
- DocumentType

Level

Page

Returns

True, if the current image is a check. Otherwise, False.

```
OCR_AConvertImage2BW("tio")  
ProcessCheckUs(true,0)
```

Parent topic: [CheckProcessingUs](#)

CMISClient actions

Content Management Interoperability Services (IBM® CMIS) is an open standard that CMISClient actions use to enable communication between Datacap applications and content management systems over the internet.

The CMISClient actions configure the connection between Datacap applications and the IBM CMIS server. You run these actions to access the CMIS server, set up document attributes and folders on the server, and upload documents to the server for storage.

- [CMISCreateFolder](#)
Creates a folder on the CMIS server.
- [CMISCreateFolderCustomType](#)
Creates a folder on the CMIS server and allows specification of the folder type.
- [CMISDeleteFile](#)
Deletes a file on the CMIS server.
- [CMISDeleteFolder](#)
Deletes a folder on the CMIS server.
- [CMISDoesFileExist](#)
Tests that a file exists on the CMIS server.
- [CMISDoesFolderExist](#)
Tests that a folder exists on the CMIS server.
- [CMISDownloadFile](#)
Download a file that is on the CMIS server to a local hard disk.
- [CMISLogDocumentTypes](#)
Logs the document types defined on the CMIS server.

- [CMISLogin](#)
Supply login credentials and connect to the CMIS server.
- [CMISRefreshClientCache](#)
Refreshes the client-side cache.
- [CMISSetDocUploadProperty](#)
Sets the value of a property that belongs to the file that is uploaded.
- [CMISSetDocUploadType](#)
Sets the type of the file that will be uploaded.
- [CMISSetVersion](#)
Sets the version type of the file that is uploaded.
- [CMISUploadFile](#)
Upload a file to the CMIS server.
- [CMISUploadPage](#)
Upload the current DCO page to the CMIS server.

Parent topic: [Global actions](#)

CMISCreateFolder

Creates a folder on the CMIS server.

Member of namespace

CMISClient

Syntax

```
bool CMISCreateFolder(string cmisFolderParentPath, string cmisFolderName)
```

Parameters

cmisFolderParentPath

Type: string

cmisFolderName

Type: string

Parameters

- cmisFolderParentPath : The path of the parent folder.
- cmisFolderName : The name of the new folder.

Smart parameters are supported for all parameters.

Returns

True if the folder is created. Otherwise, False.

Level

All levels.

Details

Creates a folder on the configured CMIS server.

Note that objects created on the CMIS server may not be immediately accessible by the client. For example, the indexing service on your CMIS server may run on a configured interval, so your new object may not be available until the indexing service completes. If you are having trouble accessing newly created objects, check the settings and documentation of your CMIS server to determine when objects should become available.

Example:

```
CMISCreateFolder ("/MyParent", "MyNewFolder")
```

Creates the folder MyNewFolder in the MyParent folder.

Parent topic: [CMISClient actions](#)

CMISCreateFolderCustomType

Creates a folder on the CMIS server and allows specification of the folder type.

Member of namespace

CMISClient

Syntax

```
bool CMISCreateFolderCustomType (string cmisFolderPath, string cmisFolderName,  
string cmisFolderType)
```

Parameters

cmisFolderPath

Type: string

cmisFolderName

Type: string

cmisFolderType

Type: string

Parameters

- cmisFolderPath: The path of the parent folder.
- cmisFolderName: The name of the new folder.
- cmisFolderType: The type of the new folder.

Smart parameters are supported for all parameters.

Returns

True if the folder is created. Otherwise, False.

Level

All levels.

Details

Creates a folder on the configured CMIS server.

Note that objects created on the CMIS server may not be immediately accessible by the client. For example, the indexing service on your CMIS server may run on a configured interval, so your new object may not be available until the indexing service completes. If you are having trouble accessing newly created objects, check the settings and documentation of your CMIS server to determine when objects should become available.

Example:

```
CMISCreateFolderCustomType("/MyParent", "MyNewFolder", "MyType")
```

Creates the folder MyNewFolder in the MyParent folder.

Parent topic: [CMISClient actions](#)

CMISDeleteFile

Deletes a file on the CMIS server.

Member of namespace

CMISClient

Syntax

```
bool CMISDeleteFile(string cmisFilePath, bool ignoreFailure)
```

Parameters

cmisFilePath
Type: string
ignoreFailure
Type: bool

Parameters

- cmisFilePath : The file to delete on the CMIS server. Smart parameters are supported.
- ignoreFailure : Set to True to ignore any failure so the action will always return True.

Returns

If ignoreFailure is True, the action always returns True, even if the file could not be deleted. If ignoreFailure is False, returns True if the file is deleted, otherwise False is returned.

Level:

All levels.

Details

Deletes the specified file in the CMIS repository.

Some, but not all, repositories have the notion of a "soft" delete, which will perform the final deletion at a later time. You should assume that once the object is deleted, that it is deleted forever. If an object is deleted, it can

only be restored if the repository provides a mechanism to restore the object.

A delete failure can be ignored by using the `ignoreFailure` parameter. This allows actions to continue even if the delete should fail.

Example:

```
CMISDeleteFile("/MyFolder/MyDoc.txt", false)
```

This returns true if `MyDoc.txt` is deleted, otherwise it returns false.

```
CMISDeleteFile("/MyFolder/MyDoc.txt", true)
```

This always returns true, even if the file could not be deleted.

Parent topic: [CMISClient actions](#)

CMISDeleteFolder

Deletes a folder on the CMIS server.

Member of namespace

CMISClient

Syntax

```
bool CMISDeleteFolder(string cmisDirectoryPath, bool ignoreFailure)
```

Parameters

`cmisDirectoryPath`

Type: string

`ignoreFailure`

Type: bool

Parameters

- `cmisDirectoryPath` : The path of the folder to delete. Smart parameters are supported.
- `ignoreFailure` : Set to True to ignore any failure so the action will always return True.

Returns

If `ignoreFailure` is True, the action always returns True, even if the folder could not be deleted. If `ignoreFailure` is False, the action will return True if the directory is deleted, otherwise False is returned.

Level

All levels.

Details

The folder must be empty for the delete to be successful. Some, but not all, CMIS repositories have the notion of a "soft" delete where the deletion is performed at a later time. You should assume that once the object is

deleted, that it is deleted forever. If an object is deleted, it can only be restored if the repository provides a mechanism to restore the object.

A delete failure can be ignored by using the `ignoreFailure` parameter. This allows actions to continue even if the delete should fail.

Example:

```
CMISDeleteFolder("/MyFolder/AnotherFolder", false)
```

This returns true if `AnotherFolder` is deleted, otherwise it returns false.

```
CMISDeleteFolder("/MyFolder/AnotherFolder", true)
```

This always returns true, even if the folder is not deleted.

Parent topic: [CMISClient actions](#)

CMISDoesFileExist

Tests that a file exists on the CMIS server.

Member of namespace

CMISClient

Syntax

```
bool CMISDoesFileExist(string cmisFilePath, bool existenceResult)
```

Parameters

`cmisFilePath`
Type: string
`existenceResult`
Type: bool

Parameters

- `cmisFilePath` : The document and path to test for existence. Smart parameters are supported.
- `existenceResult` : If True the action will return True when the file exists, otherwise False is returned. If False the action will return false if the file exists, otherwise True is returned.

Returns

`existenceResult` if True, then return True if file exists. If False, return True if file does not exist.

Level

All levels.

Details

This action tests that the provided file exists in the connected CMIS repository. This action can be configured to return true if the file exists or return true if the file does not exist.

Example:

```
CMISDoesFolderExist("/MyFolder/MyDoc.txt", True)
```

Returns true if MyDoc.txt exists in the repository and returns false if it does not exist.

```
CMISDoesFolderExist("/MyFolder/MyDoc.txt", False)
```

Returns false if MyDoc.txt exists in the repository and returns true if MyFolder does not exist.

Parent topic: [CMISClient actions](#)

CMISDoesFolderExist

Tests that a folder exists on the CMIS server.

Member of namespace

CMISClient

Syntax

```
bool CMISDoesFolderExist(string cmisDirectoryPath, bool existenceResult)
```

Parameters

cmisDirectoryPath
Type: string
existenceResult
Type: bool

Parameters

- cmisDirectoryPath : The folder and path to test for existence. Smart parameters are supported.
- existenceResult : If True the action will return True when the folder exists, otherwise False is returned. If False the action will return False if the folder exists, otherwise True is returned.

Returns

existenceResult if True, then returns True if folder exists. If False, returns True if directory does not exist.

Level

All levels.

Details

Tests that the provided directory path exists in the connected CMIS repository. This action can be configured to return true if the folder exists or return true if the folder does not exist.

Example:

```
CMISDoesFolderExist("/MyFolder/AnotherFolder", True)
```

Returns true if "AnotherFolder" exists in the repository and returns false if it does not exist.

```
CMISDoesFolderExist("/MyFolder/AnotherFolder", False)
```

Returns false if "AnotherFolder" exists in the repository and returns true if the folder does not exist.

Parent topic: [CMISClient actions](#)

CMISDownloadFile

Download a file that is on the CMIS server to a local hard disk.

Member of namespace

CMISClient

Syntax

```
bool CMISDownloadFile(string cmisFileToDownload, string fullFileNameTarget)
```

Parameters

cmisFileToDownload

Type: string

fullFileNameTarget

Type: string

Parameters

- cmisFileToDownload : The file, with the full path, to download from the CMIS repository.
- fullFileNameTarget : The path and file name for the downloaded file. Smart parameters are supported for all parameters.

Returns

Returns True, if the download is successful. Otherwise, False.

Level:

All levels.

Details

Downloads a file from the CMIS repository to a local hard disk. The file in the repository is not changed. If the target file exists, it is overwritten.

Example:

```
CMISDownloadFile("/MyCMISDir/MyCMISFile.txt", "c:\MyLocalDir\MyLocalFile.txt")
```

Downloads a copy of the file MyCMISFile.txt and names it MyLocalFile.txt on the local system.

Parent topic: [CMISClient actions](#)

CMISLogDocumentTypes

Logs the document types defined on the CMIS server.

Member of namespace

CMISClient

Syntax

```
bool CMISLogDocumentTypes ()
```

Parameters

None.

Returns

Always True.

Level

All levels.

Details

This action will query the CMIS server and log all of the server defined document types to the RRS log, providing that logging is enabled. This is a diagnostic action and is not intended for use in a production environment, nor is it supported in a production environment. CMISLogin must be called prior to this action.

It may be useful for an developer to obtain this type information during application development and has no intended use other than this limited diagnostic. Be aware that depending on the number of types and attributes defined on the CMIS server, this action may be slow and create a large log file.

Example:

```
LogDocumentTypes ()
```

Parent topic: [CMISClient actions](#)

CMISLogin

Supply login credentials and connect to the CMIS server.

Member of namespace

CMISClient

Syntax

```
bool CMISLogin (string atomPubURL, string userID, string password,  
string repositoryID)
```

Parameters

atomPubURL
Type: string
userID
Type: string
password
Type: string
repositoryID
Type: string

Parameters

- atomPubURL : The AtomPub URL for your CMIS compatible repository.
- userID : The logon user ID.
- password : The password.
- repositoryID: Optionally specify the CMIS repository ID.

Smart parameters are supported for all parameters.

Returns

True if the login was successful. Otherwise, False.

Level

All levels.

Details

This action connects to a CMIS compatible repository by using an AtomPub URL. If the repository ID is blank, the connection is made to the first repository that is returned from the CMIS connection. This action must be called before you can use any of the other CMIS actions.

Example:

```
CMISLogin("http://localhost:8080/alfresco/service/api/cmisis", "MyUserID", "@APPVARIABLES/adv/cmisispassword")
```

The password is obtained from the application service and is set by using the Application Manager.

```
CMISLogin("http://localhost:8080/alfresco/service/api/cmisis", "MyUserID", "@APPVARIABLES/adv/cmisispassword", "MyRepositoryID")
```

This example passes the optional repository ID.

Parent topic: [CMISClient actions](#)

CMISRefreshClientCache

Refreshes the client-side cache.

Member of namespace

Syntax

```
bool CMISRefreshClientCache(string milliseconds)
```

Parameters

milliseconds
Type: string

Parameters

milliseconds : Refreshes the object local cache if it is older than the specified number of milliseconds. A value of 0 refreshes immediately. A value of -1 disables this refresh feature and the CMIS interface manages the cache itself. By default, this refresh is disabled. Smart parameters are supported.

Returns

Always True.

Level

All levels.

Details

CMIS client-side caching first checks the session cache, if an object exists in the client cache. If the required object is found, that object is used without contacting the repository. It is possible that it might be a stale object.

This action refreshes the client-side cache, if it is older than the time specified in the parameter. The value of 0 refreshes the cache immediately. This action affects all subsequent CMIS actions, refreshing each object as they are used. Call this action with a value of -1 to disable this automatic refresh.

Typically this action does not need to be called. It can be called when a specific application requires a cache refresh. If this action is not called, the CMIS interface manages the cache as appropriate.

Note: Objects that are created on the CMIS server might not be immediately accessible by the client. For example, the indexing service on your CMIS server might run on a configured interval, so a new object might not be available until the indexing service completes. This refresh setting does not affect components on the CMIS server, such as the indexing service interval. If you have trouble accessing newly created objects, check the settings and documentation of your CMIS server to determine when objects become available.

Example:

```
CMISRefreshClientCache("1000")
```

Refreshes the client local cache if it is older than 1 second.

Parent topic: [CMISClient actions](#)

CMISSetDocUploadProperty

Sets the value of a property that belongs to the file that is uploaded.

Member of namespace

CMISClient

Syntax

```
bool CMISSetDocUploadProperty(string cmisDocPropertyname, string cmisDocPropertyValue,
string valueType, bool isMulti)
```

Parameters

cmisDocPropertyname

Type: string

cmisDocPropertyValue

Type: string

valueType

Type: string

isMulti

Type: bool

Parameters

- cmisDocPropertyname : The name of the CMIS property to set.
- cmisDocPropertyValue : The value of the specified property.
- valueType : The CMIS type of the value. Must be string, integer, datetime, or boolean.
- isMulti : True, if the CMIS type is a multi-value type. False, if the CMIS type is single.

Smart parameters are supported for cmisDocPropertyname, cmisDocPropertyValue and valueType.

Returns

False, if the property cannot be set or the datetime parameter cannot not be processed. Otherwise, True.

Level

All levels.

Details

This action sets a value to the specified property for files or pages that are uploaded to the CMIS repository. Use of this action is optional. Standard CMIS properties are of the form CMIS:xxx. By convention, custom CMIS properties do not use the CMIS: prefix.

This action must be called before CMISUploadFile or CMISUploadPage. If multiple properties need to be set for a document upload, call this action multiple times to set multiple properties. The property must be defined for the document type within the CMIS repository. If the property or value is not valid for the document type, the error is not reported until the upload action is called. When the upload is complete, the property settings set with this action are discarded and this action needs to be called again to set any custom properties for another upload.

For a Boolean property type, the value must be either True or False.

The datetime type accepts an input string that specifies just the date or the date and time. The application attempts to parse the input date format based on the current locale. The action uses the locale set in the Application Service / *hr_locale* variable. If that is not set, the current OS locale is used to interpret the date/time. The action attempts to ignore unrecognized data, if possible. If the month, day, or year is missing, it attempts to use the values from the current date. If a time is not specified, then the time is set to 12:00 midnight. It is recommended to provide the full short date in the correct format for the current locale, and a full 4-digit year, to reduce the chance of the action guessing values incorrectly. Only Gregorian short date formats are supported.

Example input datetime strings in the en-US locale.

- 05/01/2009 14:57:32.8 becomes 5/1/2009 2:57:32 PM
- 2009-05-01 14:57:32.8 becomes 5/1/2009 2:57:32 PM
- 2009-05-01T14:57:32.8375298-04:00 becomes 5/1/2009 11:57:32 AM
- 5/01/2008 14:57:32.80 -07:00 becomes 5/1/2008 2:57:32 PM.

Example:

```
SetCMISDocUploadType("cmisbook:poem")
SetCMISDocUploadProperty("cmisbook:author", "Edgar Allan Poe", "string", False)
SetCMISDocUploadProperty("cmisbook:Title", "The Raven", "string", False)
CMISUploadPage("MyFile.txt", "/MyCMISDir", "text/plain")
```

This example uses a predefined custom document type and then sets the author and title properties of the uploaded file.

Parent topic: [CMISClient actions](#)

CMISSetDocUploadType

Sets the type of the file that will be uploaded.

Member of namespace

CMISClient

Syntax

```
bool CMISSetDocUploadType(string cmisDocType)
```

Parameters

cmisDocType
Type: string

Parameters

cmisDocType : The document type of the uploaded file. Smart parameters are supported.

Returns

Always returns True.

Level

All levels.

Details

This action sets the CMIS document type for the file that will be subsequently uploaded in a following upload action. Use of this action is optional. If it is not called, the value of `cmis:document` is used. To set the document type, this action must be called before `CMISUploadFile` or `CMISUploadPage`.

Standard CMIS properties are of the form `CMIS:xxx`. By convention, custom CMIS properties should not use the `CMIS:` prefix. Calling this action with an empty string, will reset the value back to the default `cmis:document` type. Once the upload is complete, the document type is reset to the default value and this action will need to be called again to set a custom value for another upload.

If the specified upload type is invalid or not defined on your CMIS server, an error will occur in the upload action.

Example:

```
SetCMISDocUploadType("cmisbook:poem")
SetCMISDocUploadProperty("cmisbook:author", "Edgar Allan Poe")
SetCMISDocUploadProperty("cmisbook:poemtitle", "The Raven")
CMISUploadPage("MyFile.txt", "/MyCMISDir", "text/plain")
```

This example uses a predefined custom document type and then sets the custom author and title properties of the uploaded file.

Parent topic: [CMISClient actions](#)

CMISSetVersion

Sets the version type of the file that is uploaded.

Member of namespace

CMISClient

Syntax

```
bool CMISSetVersion(string cmisVersion)
```

Parameters

`cmisVersion`
Type: string

Parameters

`cmisSetVersion` : The version of the uploaded document. It must be one of the following options: None, Major, Minor, CheckedOut. If this action is not called before the file upload, the None type is used.

Smart parameters are supported.

Returns

Always returns True. If the specified type is invalid or if a CMIS connection is not established, the version type defaults to None.

Level

All levels.

Details

Call this action before the `CMISUploadFile` action to set the version type of the uploaded file. The support for versions depends on the target repository. Check your repository documentation to determine which version types it supports when you are using the CMIS interface. The supported types might also vary based on the specific version configuration settings of your repository. This action is optional. If this action is not called, then the version type of None is used for uploaded files.

Example:

```
CMISSetVersion("Major")
CMISUploadFile("C:\MyDir\MyFile.txt", "MyFile.txt", "/MyCMISDir", "text/plain")
```

Parent topic: [CMISClient actions](#)

CMISUploadFile

Upload a file to the CMIS server.

Member of namespace

CMISClient

Syntax

```
bool CMISUploadFile(string fullyQualifiedFileName, string cmisUploadedName,
string cmisUploadDirectory, string mimeType)
```

Parameters

fullyQualifiedFileName

Type: string

cmisUploadedName

Type: string

cmisUploadDirectory

Type: string

mimeType

Type: string

Parameters

- fullyQualifiedFileName : The file name to upload with the full path specified.
- cmisUploadedName : The wanted final name of the file in the repository.
- cmisUploadDirectory : The full path for the folder within the repository where the file is stored.
- mimeType : The mime type of the uploaded file.

Smart parameters are supported for all parameters.

Returns

True, if the file is successfully uploaded. Otherwise, False.

Level

All levels.

Details

This action uploads a file to the configured repository. You can optionally set properties of the uploaded file using the actions `CMISSetDocUploadProperty` and `CMISSetDocUploadType`.

Note: Objects that are created on the CMIS server might not be immediately accessible by the client. For example, the indexing service on your CMIS server might run on a configured interval, so your new object might not be available until the indexing service completes. If you are not able to access newly created objects, check the settings and documentation of your CMIS server to determine when objects are available.

A record of uploaded files is placed in the file `UploadRecord.xml` in the batch directory.

Example:

```
SetCMISDocUploadType("cmisbook:poem")
SetCMISDocUploadProperty("cmisbook:author", "Edgar Allan Poe")
SetCMISDocUploadProperty("cmisbook:Title", "The Raven")
CMISUploadFile("C:\MyDir\MyFile.txt", "MyFile.txt", "/MyCMISDir", "text/plain")
```

The uploaded file is placed into the `\MyCMISDir` folder and be given the mime type of `text/plain`.

Parent topic: [CMISClient actions](#)

CMISUploadPage

Upload the current DCO page to the CMIS server.

Member of namespace

CMISClient

Syntax

```
bool CMISUploadPage(string cmisUploadedName, string cmisUploadDirectory,
string mimeType)
```

Parameters

`cmisUploadedName`

Type: string

`cmisUploadDirectory`

Type: string

`mimeType`

Type: string

Parameters

- `cmisUploadedName` : The desired final name of the file in the repository.
- `cmisUploadDirectory` : The full path for the folder within the repository where the file will be stored.
- `contentType` : The mime type of the uploaded file.

Smart parameters are supported for all parameters.

Returns

True if the file is successfully uploaded. Otherwise, False.

Level

All levels.

Details

This action uploads the current DCO page to the configured repository.

Note: Objects created on the CMIS server may not be immediately accessible. For example, the indexing service on your CMIS server may run on a configured interval, so your new object may not be available until the indexing service completes. If you are having trouble accessing newly created objects, check the settings and documentation of your CMIS server to determine when objects should become available.

A record of uploaded files will be placed in the file `UploadRecord.xml` in the batch directory.

Example:

```
SetCMISDocUploadType("cmisbook:poem")
SetCMISDocUploadProperty("cmisbook:author", "Edgar Allan Poe")
SetCMISDocUploadProperty("cmisbook:poemtitle", "The Raven")
CMISUploadPage("MyFile.txt", "/MyCMISDir", "text/plain")
```

This set of actions sets property values of the page that will be uploaded, then uploads the page. The current page is placed into the `\MyCMISDir` folder and be given the mime type of `text/plain`.

Parent topic: [CMISClient actions](#)

ColorToBW actions

Use the ColorToBW actions to change the color depth of an image.

The ColorToBW actions can specify the color-to-BW conversion settings and change the color depth of an image according to these conversion settings.

- [C2BW_Convert](#)
Changes the color depth of an image according to the conversion settings specified by using `C2BW_SetAttributes`.
- [C2BW_SetAttributes](#)
Specifies the color-to-BW conversion settings.

Parent topic: [Global actions](#)

C2BW_Convert

Changes the color depth of an image according to the conversion settings specified by using `C2BW_SetAttributes`.

Syntax

```
bool C2BW_Convert(sParam)
```

Parameters

C2BW_Convert always produces a TIF image. If the source image has a TIF extension, use this parameter to provide an extension to use for a backup of the original file.

If not provided, the parameter defaults to tio. If the source image does not have a TIF extension, this parameter is ignored and use a TIF extension and the original image name will not be changed.

Smart parameters are supported.

Returns

Always True.

Level

All Levels.

Details

Changes the color depth of an image based on a set of defined attributes, creating a new TIF image.

If action C2BW_SetAttributes is not called first, a black and white image will be created.

If called at the batch level converts all images. If called at the document level, images within the document are converted. If called on the page, or field level the page is converted.

The supported input image formats are TIF (including LZW compression), PNG, BMP and JPG.

Example

```
C2BW_SetAttributes("1","0","3")
C2BW_Convert("tic")
```

Parent topic: [ColorToBW actions](#)

Related reference:

[C2BW_SetAttributes](#)

C2BW_SetAttributes

Specifies the color-to-BW conversion settings.

Syntax

```
bool C2BW_SetAttributes(string BitsPerPixel, string Palette, string Dither)
```

Parameters

string BitsPerPixel

string Palette

string Dither

Parameters

Requires 3 numeric values to configure the output image specifications:

1. Bits per pixel - 1, 4, 8, 24. A bit depth of 1 will produce a black and white image. A bit depth of 4, 8 or 24 will produce a grayscale or color image depending on the selected palette.
2. Palette - 0 Optimized, 1 Fixed, 2 Grayscale.
3. Dither - 0 None, 1 Floyd-Steinberg, 2 Ordered, 3 Optimized.

Smart parameters are supported.

Returns

Always True.

Level

All levels.

Details

Adjusts the output of the action C2BW_Convert. Optionally called before C2BW_Convert to configure the desired image output specifications. If this action is not called, these default values are used by C2BW_Convert:

- BitsPerPixel = 1 (Black and White)
- Palette = 0 (Optimized)
- Dither = 0 (None)

Example

```
C2BW_SetAttributes("1","0","3")
C2BW_Convert(tic)
```

Parent topic: [ColorToBW actions](#)

Related reference:

[C2BW_Convert](#)

Convert actions

Use the Convert actions to convert various electronic document files into TIFF image files.

The convert actions can convert files from these formats; Microsoft Excel, HTML, Microsoft Outlook, PDF, RTF, Text, Microsoft Word.

- [Common actions](#)
Use the Common actions to define the properties that are used by all of the conversion libraries for exception handling.
- [Excel actions](#)
Use the Excel actions to convert a Microsoft Excel document file into TIFF image files.
- [Html actions](#)
Use the Html actions to convert an image file from HTML format to TIFF for recognition processing by Datacap.

- [Images actions](#)
Use the Images actions to convert various image formats into a black and white TIFF image file for recognition processing by Datacap.
- [Outlook actions](#)
Use the Outlook actions to convert the text of a Microsoft Outlook message file to a TIFF image file and save any attachments to disk. Saved attachments can then be processed as needed by subsequent actions.
- [Pdf actions](#)
Use the Pdf actions to convert an image file from PDF format to TIFF for recognition processing by Datacap.
- [PdfFRE actions](#)
Use the PdfFRE actions can use the Abbyy FineReader Engine to convert an image file from PDF format to TIFF for recognition processing by Datacap.
- [Rtf actions](#)
Use the Rtf actions to convert an image file from RTF format to TIFF for recognition processing by Datacap.
- [Tiff actions](#)
Use the Tiff action to convert a multi-page TIFF image file into multiple single page TIFF image files for recognition processing by Datacap.
- [Txt actions](#)
Use the Tiff actions to convert an image file from TXT format to TIFF for recognition processing by Datacap.
- [Word actions](#)
Use the Word actions to convert an image file from Microsoft Word format to TIFF for recognition processing by Datacap.
- [Zip actions](#)
Use the Zip actions to extract the images files in a compressed ZIP archive and convert them to separate TIFF files for recognition by Datacap.

Parent topic: [Global actions](#)

Common actions

Use the Common actions to define the properties that are used by all of the conversion libraries for exception handling.

The default behavior is to abort the batch. But you can enable continuing to run the batch after a failure to increment a variable or to raise a task condition if workflow routing is configured.

Exception handling can be isolated to specific file types, with the unspecified file types aborted upon exception.

- [DeleteSourceImagePages](#)
Removes source image DCO pages.
- [ExceptionSetFileTypes](#)
Sets the file types to be monitored for exception handling.
- [ExceptionSetHandler](#)
Sets the type of exception handling to use during a batch processing failure.
- [ExceptionSetVariableName](#)
Sets the runtime document hierarchy variable to increment upon execution.
- [ExceptionSetTaskCondition](#)
Sets the task condition to be raised upon execution.
- [SetNamePattern](#)
Changes the default naming pattern for the converted files.

- [SetNamePatternFileCheck](#)
Configures file checking behavior in naming pattern for converted files.
- [PDF compliance standards](#)
For some actions that create PDF documents, you can set the compliance standard for the produced PDF document.

Parent topic: [Convert actions](#)

DeleteSourceImagePages

Removes source image DCO pages.

Member of namespace

Convert

Syntax

```
bool DeleteSourceImagePages ()
```

Parameters

None

Returns

True, if the action is successful.

False, if an error is encountered.

Level

Batch.

Details

Removes pages where the associated file has been converted using the convert library actions. This action evaluates all the documents and pages, which are children of the batch folder. It deletes the pages that have generated new pages as a result of the convert actions.

Note: Pages are only deleted from the DCO but not the batch directory

Example:

```
DeleteSourceImagePages ()
```

Parent topic: [Common actions](#)

ExceptionSetFileTypes

Sets the file types to be monitored for exception handling.

Member of namespace

Convert

Syntax

```
bool ExceptionSetFileTypes(string types)
```

Parameters

types

Type: string

A comma delimited list of the file extensions to monitor.

Returns

Always True.

Level

Any level.

Details

Sets one or more file types to be monitored for special exception handling. Unspecified file types that encounter processing failures cause an abort without any further action. If this action is not called or the parameter is blank, then all file types are monitored.

Example:

```
ExceptionSetHandler(1)  
ExceptionSetFileTypes(.pdf, .doc, .docx, .xls, .xlsx, .txt)
```

Parent topic: [Common actions](#)

Related reference:

[ExceptionSetHandler](#)

ExceptionSetHandler

Sets the type of exception handling to use during a batch processing failure.

Member of namespace

Convert

Syntax

```
bool ExceptionSetHandler(int handler)
```

Parameters

handler

Type: integer

The type of exception handler.

Parameters

- 0: Abort the batch (Default).
- 1: Increment a variable in the runtime hierarchy, do not abort the batch.
- 2: Raise a task condition, do not abort the batch.

Returns

True, if a valid parameter is specified. Otherwise, False.

Level

Any level.

Details

Use this action to change or reset the abort behavior that is caused by conversion failures. Avoiding aborts can be useful when an external workflow process is already in place.

Example:

```
ExceptionHandler(0)
```

Parent topic: [Common actions](#)

Related reference:

[ExceptionHandlerFileTypes](#)

[ExceptionHandlerVariableName](#)

[ExceptionHandlerTaskCondition](#)

ExceptionHandlerVariableName

Sets the runtime document hierarchy variable to increment upon execution.

Member of namespace

Convert

Syntax

```
bool ExceptionSetVariableName(string varName)
```

Parameters

varName

Type: string

The variable to increment. Smart parameters are supported.

Returns

Always True.

Level

Any level.

Details

Sets the variable to increment upon execution.

The default value is `@B.ConvertExceptions` when the `ExceptionHandler` action is set to use the `var` variable value.

Example:

```
ExceptionHandler(1)
ExceptionSetVariableName(@B.PDFConvertExceptions)
```

Parent topic: [Common actions](#)

Related reference:
[ExceptionHandler](#)

ExceptionSetTaskCondition

Sets the task condition to be raised upon execution.

Member of namespace

Convert

Syntax

```
bool ExceptionSetTaskCondition(int taskCondition)
```

Parameters

`taskCondition`
Type: integer
A zero-based task condition index.

Returns

Always True.

Level

Any level.

Details

Sets the task condition, which is defined in workflow administration, to be raised upon execution.

Batch splitting is not supported.

Example:

```
ExceptionHandler(2)
ExceptionSetTaskCondition(0)
```

Parent topic: [Common actions](#)

Related reference:
[ExceptionHandler](#)

SetNamePattern

Changes the default naming pattern for the converted files.

Member of namespace

Convert

Syntax

```
bool SetNamePattern(string PatternType)
```

Parameters

PatternType

Type: string

A single positive numeric value.

Parameters

Value of 1 or 2.

- 1: Uses the default AlphaDecimal pattern, which consists of 4 pairs of 2 characters from 01 to ZZ.
- 2: Uses a TMxxxxxx pattern where xxxxxx is a number from 000001 to 999999.

Returns

True, if the setting is successful.

False, if the setting is rejected.

Level

Any level.

Details

Changes the default naming pattern for the converted files.

Example:

```
SetNamePattern(2)
```

Parent topic: [Common actions](#)

SetNamePatternFileCheck

Configures file checking behavior in naming pattern for converted files.

Member of namespace

Convert

Syntax

```
bool SetNamePatternFileCheck (bool fileCheck)
```

Parameters

- True - Enable file checking, naming pattern will increment count if preexisting files are found (Default).
- False - Disable file checking, preexisting files may be overwritten.

Returns

Always True.

Level

All.

Details

This action only applies to the AlphaDecimal naming pattern. Use this action to avoid appending files to the batch directory. When file checking is enabled, new output files do not overwrite the preexisting files in scenarios such as batch reprocessing, leading to increased disk consumption.

Example:

```
SetNamePatternFileCheck (False)
```

Parent topic: [Common actions](#)

PDF compliance standards

For some actions that create PDF documents, you can set the compliance standard for the produced PDF document.

Set the value of the variable `targetPdfCompliance` in accordance with the following table. This variable is on the DCO objects for which you are calling the action. The variable's value must be set before you call the action.

Standard	targetPdfCompliance value
Pdf15	No value is set. (This standard is the default standard.)
PdfA1a	"1"
PdfA1b	"2"

Parent topic: [Common actions](#)

Excel actions

Use the Excel actions to convert a Microsoft Excel document file into TIFF image files.

The Excel actions can resize columns or rows from a table, set page orientation, create blank pages, and print grid lines when you convert documents from Excel.

Restriction: The actions in the following table work with files from Excel 2000 or later.

- [ExcelAutoFitColumns](#)
Sets the automatic sizing of all columns for Excel Workbook converted to TIFF.
- [ExcelAutoFitRows](#)
Sets the automatic sizing of all rows for Excel Workbook converted to TIFF.
- [ExcelOrientationToLandscape](#)
Forces the orientation of Excel files to landscape for ExcelWorkbookToImage.
- [ExcelOrientationToPortrait](#)
Forces the orientation of Excel files to portrait for ExcelWorkbookToImage.
- [ExcelPrintBlankPage](#)
Determines if blank pages are created when converting Excel Workbook to TIFF.
- [ExcelPrintGridlines](#)
Enables or disables gridlines when converting Excel files to TIFF.
- [ExcelPrintQuality](#)
Adjusts the resolution of the image output by ExcelWorkbookToImage.
- [ExcelScalingFactor](#)
Forces the print scaling of Excel Workbook to a specific value for ExcelWorkbookToImage.
- [ExcelShapeMinArea](#)
Specifies the minimum area required for shape rendering for ExcelWorkbookToImage.
- [ExcelTiffCompression](#)
Sets the compression used in the TIFF output by ExcelWorkbookToImage.
- [ExcelWorkbookToImage](#)
Converts a page with *.xls or *.xlsx file to a page or pages in TIFF format.
- [ExcelWorkbookToImageEx](#)
Converts a page with *.xls or *.xlsx file to a page or pages in TIFF format.
- [ExcelWorkbookToPdf](#)
Converts .xls or .xlsx files to PDF document format.

Parent topic: [Convert actions](#)

ExcelAutoFitColumns

Sets the automatic sizing of all columns for Excel Workbook converted to TIFF.

Member of namespace

Convert

Syntax

```
bool ExcelAutoFitColumns (bool autoFitColumns)
```

Parameters

autoFitColumns
Type: bool

Parameters

autoFitColumns : A Boolean value that enables and disables the automatic sizing of columns in an Excel file.

True: The columns are set to automatically adjust the size based on the content.

False: The columns do not have their size adjusted.

Returns

Always True.

Level

Page level.

Details

This enables the auto sizing of columns to shrink or increase the size of the column to fit all of the data within the column. This action must be called before `ExcelWorkbookToImage`. If this action is not called, the setting that was saved in the original Excel file is used.

Example:

```
ExcelAutoFitColumns ("False")
ExcelWorkbookToImage ()
```

Parent topic: [Excel actions](#)

ExcelAutoFitRows

Sets the automatic sizing of all rows for Excel Workbook converted to TIFF.

Member of namespace

Convert

Syntax

```
bool ExcelAutoFitRows (bool autoFitRows)
```

Parameters

autoFitRows
Type: bool

Parameters

autoFitRows : A Boolean value that enables and disables the automatic sizing of rows in an Excel file.

True: The rows are set to automatically adjust the size based on the content.

False: The rows do not have their size adjusted.

Returns

Always True.

Level

Page level.

Details

This enables the auto sizing of rows to shrink or increase the size of the row to fit all of the data within the row. This action must be called before `ExcelWorkbookToImage`. If this action is not called, the setting that was saved in the original Excel file will be used.

Example:

```
ExcelAutoFitRows ("False")
ExcelWorkbookToImage ()
```

Parent topic: [Excel actions](#)

ExcelOrientationToLandscape

Forces the orientation of Excel files to landscape for `ExcelWorkbookToImage`.

Member of namespace

Convert

Syntax

```
bool ExcelOrientationToLandscape ()
```

Parameters

None.

Returns

Always True.

Level

Page level.

Details

The TIFF files created from `ExcelWorkbookToImage` will all be created with a landscape orientation. This action must be called before `ExcelWorkbookToImage`. If this action is not called, the portrait / landscape setting that was saved in the original Excel file is used.

Example:

```
ExcelOrientationToLandscape ()
ExcelWorkbookToImage ()
```

Parent topic: [Excel actions](#)

ExcelOrientationToPortrait

Forces the orientation of Excel files to portrait for ExcelWorkbookToImage.

Member of namespace

Convert

Syntax

```
bool ExcelOrientationToPortrait ()
```

Parameters

None.

Returns

Always True.

Level

Page level.

Details

The TIFF files created from ExcelWorkbookToImage will all be created with a portrait orientation. This action must be called before ExcelWorkbookToImage. If this action is not called, the portrait / landscape setting that was saved in the original Excel file is used.

Example:

```
ExcelOrientationToPortrait()  
ExcelWorkbookToImage()
```

Parent topic: [Excel actions](#)

ExcelPrintBlankPage

Determines if blank pages are created when converting Excel Workbook to TIFF.

Member of namespace

Convert

Syntax

```
bool ExcelPrintBlankPage (bool blankPage)
```

Parameters

blankPage

Type: bool

Parameters

blankPage : A Boolean value that enables or disables creation of blank TIFFs when there is a blank Excel page.

True: A blank TIFF is created if the page is blank.

False: A blank TIFF is not created if the page is blank.

Returns

Always True.

Level

Page level.

Details

When printing an Excel Workbook, it is typical to have overflow pages that sometimes have data and sometimes are blank. Setting the value to true creates a TIFF for the blank overflow pages in this process. If this action is not called before ExcelWorkbookToImage, then the default of False is used.

Example:

```
ExcelPrintBlankPage ("False")  
ExcelWorkbookToImage ()
```

Parent topic: [Excel actions](#)

ExcelPrintGridlines

Enables or disables gridlines when converting Excel files to TIFF.

Member of namespace

Convert

Syntax

```
bool ExcelPrintGridlines (bool gridlines)
```

Parameters

gridlines

Type: bool

Parameters

gridlines : A Boolean value that determines if gridlines are displayed in Excel files converted to TIFF.

True: Gridlines are included in the converted image.

False: Gridlines are not shown in the converted image.

Returns

Always True.

Level

Page level.

Details

The TIFF files created from `ExcelWorkbookToImage` will all use the specified grid setting when converted to TIFF. This action must be called before `ExcelWorkbookToImage`. If this action is not called, the grid line setting that was saved in the original Excel file is used.

Example:

```
ExcelPrintGridlines("False")
ExcelWorkbookToImage()
```

Parent topic: [Excel actions](#)

ExcelPrintQuality

Adjusts the resolution of the image output by `ExcelWorkbookToImage`.

Member of namespace

Convert

Syntax

```
bool ExcelPrintQuality (int dpi)
```

Parameters

dpi
Type: int

Parameters

dpi : A single positive numeric value for the dots per inch (dpi) of the output image.

Returns

False if the parameter is invalid. Otherwise, True.

Level

Page level.

Details

Sets the resolution of the output image for `ExcelWorkbookToImage`. If this action is not called, the default value of 200 dpi will be used. Typically, input documents for recognition are 200 dpi.

Example:

```
ExcelPrintQuality(200)
ExcelWorkbookToImage()
```

Parent topic: [Excel actions](#)

ExcelScalingFactor

Forces the print scaling of Excel Workbook to a specific value for `ExcelWorkbookToImage`.

Member of namespace

Convert

Syntax

```
bool ExcelScalingFactor (int percent)
```

Parameters

percent
Type: int

Parameters

percent: A positive integer that controls the scaling by percentage used when converting an Excel Workbook to TIFF. For example, a value of 100 means to print at 100%.

Returns

Always True.

Level

Page level.

Details

The TIFF files created from `ExcelWorkbookToImage` will all use the specified printing scale value when converted to TIFF. This action must be called before `ExcelWorkbookToImage`. If this action is not called, the scaling factor that was saved in the original Excel file is used.

Example:

```
ExcelScalingFactor(100)
ExcelWorkbookToImage()
```

Parent topic: [Excel actions](#)

ExcelShapeMinArea

Specifies the minimum area required for shape rendering for ExcelWorkbookToImage.

Member of namespace

Convert

Syntax

```
bool ExcelShapeMinArea (long minShapeArea)
```

Parameters

minShapeArea -1 to render all shapes, 0 to remove all shapes, or a positive pixel value. Default is -1.

Returns

Always True.

Level

Page level.

Details

Sets the minimum pixel area required by shape objects in a workbook. When a positive numeric value is specified, any shape whose area is less than the parameter value will not be output during ExcelWorkBookToImage. Shapes include object types such as lines, rectangles, and charts.

Example:

```
ExcelShapeMinArea (100)  
ExcelWorkbookToImage ()
```

Parent topic: [Excel actions](#)

ExcelTiffCompression

Sets the compression used in the TIFF output by ExcelWorkbookToImage.

Member of namespace

Convert

Syntax

```
bool ExcelTiffCompression (string tiffCompression)
```

Parameters

tiffCompression
Type: string

Parameters

tiffCompression is one of the following values to set the TIFF compression:

- NONE : A color image with no compression.
- LZW : A color image using LZW compression.
- CCITT4 : A black and white image with fax CCITT4 compression.

Returns

Always True.

Level

Page level.

Details

Sets the compression of the output image from ExcelWorkbookToImage. If this action is not called, the default value of CCITT4 is used. Typically, input documents for recognition are black and white with fax compression.

Example:

```
ExcelTiffCompression("CCITT4")
ExcelWorkbookToImage()
```

Parent topic: [Excel actions](#)

ExcelWorkbookToImage

Converts a page with *.xls or *.xlsx file to a page or pages in TIFF format.

Member of namespace

Convert

Syntax

```
bool ExcelWorkbookToImage ()
```

Parameters

None.

Returns

True if the file is successfully converted to a TIFF document.

False if the current page is not an Excel Workbook or if there is a failure in the conversion.

If the number of input files/pages exceeds the maximum allowed or if there is a failure in the conversion, the batch is set to abort.

Level

Page level.

Details

If the current page is an Excel Workbook, the file is converted to multiple TIFF files, one TIFF file for each page within the Workbook, based on the settings of the other Excel actions that configure the conversion settings.

Each new TIFF also has a new page created within the application environment which can be processed by subsequent rules. The original file name from which the page was extracted is stored in the *ParentImage* variable, for possible future reference within your application.

If the configured output image format and compression only supports black and white, such as CCITT4, colored text is exported as black and background shading is set to white.

Example:

```
ExcelPrintQuality(200)
ExcelTiffCompression(CCITT4)
ExcelPrintBlankPage(False)
ExcelWorkbookToImage()
```

Parent topic: [Excel actions](#)

ExcelWorkbookToImageEx

Converts a page with *.xls or *.xlsx file to a page or pages in TIFF format.

Member of namespace

Convert

Syntax

```
bool ExcelWorkbookToImageEx (int maxOutputPages, int startingSheet, bool
singleSheetOutput)
```

Parameters

maxOutputPages

0 to convert all pages within the spreadsheet, or a positive integer value. Default is 0.

startingSheet

Zero-based sheet index denoting the starting point of conversion. Default is 0.

singleSheetOutput

True to only convert the starting sheet regardless of how many pages are allowed by maxOutputPages. Default is false.

Returns

True if the file is successfully converted to a TIFF document.

False if the current page is not an Excel Workbook or if there is a failure in the conversion.

If the number of input files/pages exceeds the maximum allowed or if there is a failure in the conversion, the batch is set to abort.

Level

Page level.

Details

If the current page is an Excel Workbook, the file is converted to multiple TIFF files, one TIFF file for each page within the Workbook, based on the settings of the other Excel actions that configure the conversion settings.

The number of converted pages can be capped to avoid excessive output. You can also specify the starting sheet index. If the value is out of range then the last sheet will be used. Conversion is limited to a single sheet when specifically enabled (by using `singleSheetOutput` or when the maximum converted page count (`maxOutputPages`) has been met.

Each new TIFF also has a new page created within the application environment which can be processed by subsequent rules. The original file name from which the page was extracted is stored in the *ParentImage* variable, for possible future reference within your application.

If the configured output image format and compression only supports black and white, such as CCITT4, colored text is exported as black and background shading is set to white.

Example:

```
ExcelPrintQuality(200)
ExcelTiffCompression(CCITT4)
ExcelPrintBlankPage(False)
ExcelWorkbookToImageEx(5,0,false)
```

Parent topic: [Excel actions](#)

ExcelWorkbookToPdf

Converts .xls or .xlsx files to PDF document format.

Member of namespace

Convert

Syntax

```
bool ExcelWorkbookToPdf ()
```

Returns

True if the file conversion succeeds. Otherwise, the action returns False if the current page is not an Excel document or some other conversion failure occurs..

Level

Page level.

Details

Use this action to convert Excel files to a single PDF file. To set the compliance standard for the produced PDF document, set the value of the variable `targetPdfCompliance` in accordance with the following table:

Standard	targetPdfCompliance value
Pdf15	No value is set. (This standard is the default standard.)
PdfA1b	"1"

This action sets the batch to abort in the following circumstances:

- The number of input files or pages exceeds the allowed maximum.
- The conversion fails.

Example:

```
ExcelWorkbookToPdf
```

Parent topic: [Excel actions](#)

Html actions

Use the Html actions to convert an image file from HTML format to TIFF for recognition processing by Datacap.

The Html actions can specify output resolution and the compression algorithm to use when you convert documents from HTML to TIFF.

- [HtmlLayout](#)
Sets the custom layout settings that the `HtmlToImage` action uses for converting HTML pages to TIFF format.
- [HtmlPrintQuality](#)
Adjusts the resolution of the image output by `HtmlToImage`.
- [HtmlTiffCompression](#)
Sets the compression used in the TIFF output by `HtmlToImage`.
- [HtmlToImage](#)
Converts a page with `*.htm` or `*.html` file to a page or pages in TIFF format.
- [HtmlToPdf](#)
Converts `.htm` or `.html` files to PDF document format.

Parent topic: [Convert actions](#)

HtmlLayout

Sets the custom layout settings that the `HtmlToImage` action uses for converting HTML pages to TIFF format.

Member of namespace

Convert

Syntax

```
bool HtmlLayout(string sMargins, int tableFit, bool tableBorders, bool cellShading,  
bool
```

clearBackground)

Parameters

sMargins

A comma-delimited list of the left, top, right, and bottom margin sizes. Each item on the list is a positive integer that represents the margin size in inches multiplied by 10. For example, if the passed parameter value is “6,6,6,6”, all margin sizes are 0.6 inches. The default margin size is 0.5 inches.

tableFit

The conversion method to use for HTML tables. The method is specified by one of the following values:

0	Auto fit: Shrinks cells according to their contents.
1	Available space: Expands cells to the available page width.
2	Fixed width: Sizes cells according to their width properties.

The default method is auto fit.

tableBorders

Whether to render table borders. The following values are possible:

True	Table borders are rendered. Important: Rendering borders might cause issues later during CCO normalization, such as misbehavior during text location, click and key, and other operations. For more information about CCO normalization, see NormalizeCCO .
False	Table borders are not rendered.

The default value is False.

cellShading

Whether to render table cell shading. The following values are possible:

True	Table cells are rendered with shading. Important: Shading can have the following adverse effects: <ul style="list-style-type: none">• Speckling. Speckling can interfere with character recognition, fingerprint matching, and other operations.• Unreadable text. Unreadable text can occur for bitonal table cells that have background noise.
False	Table cells are not rendered with shading.

For color output, the default value is True. For all other output, the default value is False.

clearBackground

Whether to clear the background of the HTML page. The following values are possible:

True	The background is cleared. Doing so might improve the clarity of color output.
False	The background is not cleared.

The default value is False.

Returns

Always True.

Level

Page level.

Details

Example:

```
HtmlLayout ("25,25,25,25", 1, False, True, True)
HtmlToImage ()
```

Parent topic: [Html actions](#)

Related reference:

[HtmlToImage](#)

HtmlPrintQuality

Adjusts the resolution of the image output by `HtmlToImage`.

Member of namespace

Convert

Syntax

```
bool HtmlPrintQuality (int dpi)
```

Parameters

dpi

Type: int

A single positive numeric value for the dots per inch (dpi) of the output image.

Returns

False if the parameter is invalid. Otherwise, True.

Level

Page level.

Details

Sets the resolution of the output image for `HtmlToImage`. If this action is not called, the default value of 200 dpi is used. Typically, input documents for recognition are 200 dpi.

Example:

```
HtmlPrintQuality (200)
HtmlDocumentToImage ()
```

Parent topic: [Html actions](#)

HtmlTiffCompression

Sets the compression used in the TIFF output by HtmlToImage.

Member of namespace

Convert

Syntax

```
bool HtmlTiffCompression (string tiffCompression)
```

Parameters

tiffCompression

Type: string

A parameter of one of the following values to set the TIFF compression:

Parameters

- NONE : A color image with no compression.
- LZW : A color image using LZW compression.
- CCITT4 : A black and white image with fax CCITT4 compression.

Returns

Always True.

Level

Page level.

Details

Sets the compression of the output image from HtmlToImage. If this action is not called, the default value of CCITT4 is used. Typically, input documents for recognition are black and white with fax compression.

Example:

```
HtmlTiffCompression (CCITT4)  
HtmlToImage ()
```

Parent topic: [Html actions](#)

HtmlToImage

Converts a page with *.htm or *.html file to a page or pages in TIFF format.

Member of namespace

Convert

Syntax

```
bool HtmlToImage ()
```

Parameters

None.

Returns

True if the file is successfully converted to a TIFF document.

False if the current page is not an Html Document or if there is a failure in the conversion.

If the number of input files/pages exceeds the maximum allowed or if there is a failure in the conversion, the batch will be set to abort.

Level

Page level.

Details

If the current page is an Html Document, the file is converted to multiple TIFF files, one TIFF file for each page within the Document, based on the settings of the other Html actions that configure the conversion settings.

Each new TIFF also has a new page created within the application environment which can be processed by subsequent rules. The original file name from which the page was extracted is stored in the ParentImage variable, for possible future reference within your application.

If the configured output image format and compression only supports black and white, such as CCITT4, colored text is exported as black. Html table borders will not be rendered onto output images.

Example:

```
HtmlPrintQuality(200)
HtmlTiffCompression(CCITT4)
HtmlToImage()
```

Parent topic: [Html actions](#)

HtmlToPdf

Converts .htm or .html files to PDF document format.

Member of namespace

Convert

Syntax

```
bool HtmlToPdf ()
```

Returns

True if the file conversion succeeds. Otherwise, the action returns False if the current page is not an HTML document or some other conversion failure occurs..

Level

Page level.

Details

Use this action to convert HTML files to a single PDF file. For information about setting the compliance standard for the produced PDF document, see [PDF compliance standards](#).

This action sets the batch to abort in the following circumstances:

- The number of input files or pages exceeds the allowed maximum.
- The conversion fails.

Example:

```
HtmlToPdf
```

Parent topic: [Html actions](#)

Images actions

Use the Images actions to convert various image formats into a black and white TIFF image file for recognition processing by Datacap.

The Images actions can specify default DPI settings and image type file extensions for the images that you want to convert to TIFF.

- [ImageDefaultDPI](#)
Sets the default dpi (dots per inch) when converting images that do not have an embedded dpi value.
- [ImageFileTypesToConvert](#)
Sets the file extension values of image types to convert to tiff.
- [ImageMonoThreshold](#)
Sets the threshold value when converting Image to 1 bit tiff using threshold type conversion.
- [ImageMonoType](#)
Sets the method to use when converting color images to black and white tiffs.
- [ImageToTIFF](#)
Converts an Image file to TIFF format.

Parent topic: [Convert actions](#)

ImageDefaultDPI

Sets the default dpi (dots per inch) when converting images that do not have an embedded dpi value.

Member of namespace

Convert

Syntax

```
bool ImageDefaultDPI (short X, short Y)
```

Parameters

X

Type: short

Horizontal dpi (X axis). A positive numeric value normally ranging from 96 to 300

Y

Type: short

Vertical dpi (Y axis). A positive numeric value normally ranging from 96 to 300

Returns

False if there is a failure to set the default X or Y dpi value. Otherwise True.

Level

Page level.

Details

Sets the default dpi to use when converting color or grayscale images to TIFF when the source image does not have a dpi value.

Example:

```
ImageFileTypesToConvert (".jpg, .jpeg, gif, bmp")
ImageDefaultDPI ("200", "200")
ImageMonoType (4)
ImageMonoThreshold (9)
ImageToTIFF ()
```

Parent topic: [Images actions](#)

ImageFileTypesToConvert

Sets the file extension values of image types to convert to tiff.

Member of namespace

Convert

Syntax

```
bool ImageFileTypesToConvert (string fileextensions)
```

Parameters

fileextensions

Type: string

A CSV string of file extensions that defines the image types that will be converted.

Parameters

fileextensions : A CSV string of file extensions that defines the image types that will be converted.

Returns

Always True.

Level

Page level.

Details

Sets the file extension values of image types to convert to TIFF. The file types of JPEG, BMP, PNG, TIFF and GIF are supported. A period prefixing the extension is allowed, but not required. Because TIFF is supported, it is possible to use this action to convert a color TIFF to a Black and White TIFF.

This action must be used prior to ImageToTIFF.

Example:

```
ImageFileTypesToConvert (".jpg, .jpeg, gif, bmp")
ImageToTIFF ()
```

Parent topic: [Images actions](#)

ImageMonoThreshold

Sets the threshold value when converting Image to 1 bit tiff using threshold type conversion.

Member of namespace

Convert

Syntax

```
bool ImageMonoThreshold (short thresh)
```

Parameters

thresh

Type: short

A positive numeric value from 1 to 255.

Parameters

thresh: A positive numeric value from 1 to 255.

Returns

Always True.

Level

Page level.

Details

Sets the B/W conversion algorithm value to use when converting color or grey scale images to TIFF with the Threshold method option.

Example:

```
ImageFileTypesToConvert (".jpg, .jpeg, gif, bmp")
ImageMonoType (4)
ImageMonoThreshold (9)
ImageToTIFF ()
```

Parent topic: [Images actions](#)

ImageMonoType

Sets the method to use when converting color images to black and white tiffs.

Member of namespace

Convert

Syntax

```
bool ImageMonoType (int Mono)
```

Parameters

Mono
Type: int

Parameters

Mono : A positive numeric value of 1 to 4 for the type of conversion to use.

1. Convert image using Diffusion method.
2. Convert image using Halftone method.
3. Convert image using Bayer method.
4. Convert image using Threshold method with a threshold value. Value defaults to 10.

Returns

Always True.

Level

Page level.

Details

Sets the black and white conversion algorithm to use when converting color or grayscale images to TIFF. If you are using the threshold method, you must also call the ImageMonoThreshold action prior to converting the image with ImageToTIFF.

Error diffusion is a type of halftoning in which the quantization residual is distributed to neighboring pixels that have not yet been processed. Its main use is to convert a color image into a black and white image. Halftone is

the reprographic technique that simulates continuous tone imagery through the use of dots, varying either in size or in spacing. Bayer method is an image dithering algorithm that is commonly used to maintain a the characteristics of a photo image of higher colors in an image of less color depth. The threshold method makes individual pixels in an image black if their value is greater than the threshold value and the remaining pixels white.

For best results, you may need to experiment with the different types of images that you are expecting to process and select the conversion method that gives you the best results. If the resulting TIFF images are going to be used in a subsequent recognition process, pick the technique that works best to output dark, solid characters with as little background noise, or dithering around the characters, as possible.

Example:

```
ImageFileTypesToConvert(".jpg, .jpeg, gif, bmp")
ImageMonoType(1)
ImageToTIFF()
```

Parent topic: [Images actions](#)

ImageToTIFF

Converts an Image file to TIFF format.

Member of namespace

Convert

Syntax

```
bool ImageToTIFF ()
```

Parameters

None.

Returns

True if the file is successfully converted to a TIFF document.

False if the current page is not a supported Image type or if there is a failure in the conversion.

If the number of input files/pages exceeds the maximum allowed or if there is a failure in the conversion, the batch is set to abort.

Level

Page level.

Details

If the current page is an Image, the file will be converted to a single TIFF file. ImageFileTypesToConvert must be used along with ImageToTIFF. Only the image types set by a previous call to ImageFileTypesToConvert is converted to a TIFF. No images are converted if ImageFileTypesToConvert has not been set.

Each TIFF will also have a new page created within the application environment which can be processed by subsequent rules. The original file name from which the page was extracted is stored in the *ParentImage* variable, for possible future reference within your application.

Example:

```
ImageFileTypesToConvert (".jpg, .jpeg, gif, bmp")
ImageToTIFF ()
```

Parent topic: [Images actions](#)

Outlook actions

Use the Outlook actions to convert the text of a Microsoft Outlook message file to a TIFF image file and save any attachments to disk. Saved attachments can then be processed as needed by subsequent actions.

The Outlook actions can specify the image resolution and the compression algorithm that is used when you convert from Outlook to TIFF.

- [OutlookAttachmentTypeIndicator](#)
Determines if an email attachment type is set based on the extension or mime type.
- [OutlookMessageToAttachmentOnly](#)
Extracts attachments from *.msg or *.eml to pages without converting the MSG to a TIFF.
- [OutlookMessageToImageAndAttachment](#)
Converts a *.msg or *.eml file to a page or pages in TIFF format.
- [OutlookPrintQuality](#)
Adjusts the resolution of the image output by OutlookMessageToImageAndAttachment.
- [OutlookTiffCompression](#)
Sets the compression used in the TIFF output by OutlookMessageToImageAndAttachment.

Parent topic: [Convert actions](#)

OutlookAttachmentTypeIndicator

Determines if an email attachment type is set based on the extension or mime type.

Member of namespace

Convert

Syntax

```
bool OutlookAttachmentTypeIndicator (string typeIndicator)
```

Parameters

- MIME – The attachment type is determined by the email mime type.
- EXT – The attachment type is determined by the attachment extension.

Smart parameters are supported.

Returns

Always True.

Level

Any level.

Details

When you save e-mail attachments to a file, the file type can be set based on the mime type specified in the e-mail or based on the actual extension of the attachment. This action is useful when you receive emails which may sometimes have an incorrect extension or incorrect mime type, allowing the application to name the extension based on the setting. If the primary setting is blank, then the alternative method is used. For example, if the extension is set as the primary indicator and the attachment has a blank extension, then the mime type is used. If this action is not called, or if the provided `typeIndicator` parameter is incorrect, then the setting is default to "MIME" and saved attachments are named based on the mime type indicated in the e-mail. This action needs to be called prior to the action that performs the attachment extraction. In scenarios where the application needs to processes e-mails that have incorrect extensions, this action can be used to set the mime type as the indicator for the attachment file type. Conversely, if the application needs to process emails that have mime types set incorrectly, then the attachment extension can be set to the primary indicator for the file type extension.

Example:

```
OutlookAttachmentTypeIndicator ("EXT")
```

This example uses the extension of the attachment to determine the file type of the attachment saved to disk. If the attachment is named "MyFile.PDF" and the mime type is set to text/plain, then the extracted attachment has a PDF extension.

Parent topic: [Outlook actions](#)

OutlookMessageToAttachmentOnly

Extracts attachments from *.msg or *.eml to pages without converting the MSG to a TIFF.

Member of namespace

Convert

Syntax

```
bool OutlookMessageToAttachmentOnly ()
```

Parameters

None.

Returns

True, if the attachments are successfully extracted from the message, or if the message contains no attachments.

False, if the current page is not an Outlook Message or if there is a failure in the conversion.

If the number of input files/pages exceeds the maximum that is allowed or if there is a failure in the conversion, the batch is set to `abort`.

Level

Page level.

Details

If the current page is an Outlook Message, the attachments within the message are saved as separate files.

Each attachment or embedded image within the email is removed from the email and placed on disk. A new page entry is created for each attachment that can be processed by subsequent rules. The original file name from which the page was extracted will be stored in the *ParentImage* variable, for possible future reference within your application.

Example:

```
OutlookMessageToAttachmentOnly()
```

Parent topic: [Outlook actions](#)

Related reference:

[OutlookMessageToImageAndAttachment](#)

OutlookMessageToImageAndAttachment

Converts a *.msg or *.eml file to a page or pages in TIFF format.

Member of namespace

Convert

Syntax

```
bool OutlookMessageToImageAndAttachment ()
```

Parameters

None.

Returns

True if the file is successfully converted to a TIFF document.

False if the current page is not an Outlook Message or if there is a failure in the conversion.

If the number of input files/pages exceeds the maximum allowed or if there is a failure in the conversion, the batch is set to abort.

Level

Page level.

Details

If the current page is an Outlook Message, the file is converted to multiple TIFF files, one TIFF file for each page within the Message, based on the settings of the other Outlook actions that configure the conversion settings. A copy of the original MSG file without attachments is created and converted to TIFF as well.

Each new TIFF also has a new page created within the application environment which can be processed by subsequent rules. The original file name from which the page was extracted is stored in the *ParentImage* variable, for possible future reference within your application.

Each attachment or embedded image within the e-mail is removed from the e-mail and placed on disk. A new page entry will be created for each attachment that can be processed by rules.

If the configured output image format and compression only supports black and white, such as CCITT4, colored text is exported as black.

Example:

```
OutlookPrintQuality(200)
OutlookTiffCompression("CCITT4")
OutlookMessageToImageAndAttachment()
```

Parent topic: [Outlook actions](#)

OutlookPrintQuality

Adjusts the resolution of the image output by OutlookMessageToImageAndAttachment.

Member of namespace

Convert

Syntax

```
bool OutlookPrintQuality (int dpi)
```

Parameters

dpi
Type: int

Parameters

dpi : A single positive numeric value for the dots per inch (dpi) of the output image.

Returns

False if the parameter is invalid. Otherwise, True.

Level

Page level.

Details

Sets the resolution of the output image for OutlookMessageToImageAndAttachment. If this action is not called, the default value of 200 dpi is used. Typically, input documents for recognition are 200 dpi.

Example:

```
OutlookPrintQuality(200)
OutlookMessageToImageAndAttachment()
```

Parent topic: [Outlook actions](#)

OutlookTiffCompression

Sets the compression used in the TIFF output by OutlookMessageToImageAndAttachment.

Member of namespace

Convert

Syntax

```
bool OutlookTiffCompression (string tiffCompression)
```

Parameters

tiffCompression
Type: string

Parameters

tiffCompression : A parameter of one of the following values to set the TIFF compression:

- NONE : A color image with no compression.
- LZW : A color image using LZW compression.
- CCITT4 : A black and white image with fax CCITT4 compression.

Returns

Always True.

Level

Page level.

Details

Sets the compression of the output image from OutlookMessageToImageAndAttachment. If this action is not called, the default value of CCITT4 is used. Typically, input documents for recognition are black and white with fax compression.

Example:

```
OutlookTiffCompression("CCITT4")
OutlookMessageToImageAndAttachment()
```

Parent topic: [Outlook actions](#)

Pdf actions

Use the Pdf actions to convert an image file from PDF format to TIFF for recognition processing by Datacap.

The Pdf actions can specify bit depth of the output image, as well as the compression algorithm and conversion method to use when you convert from PDF to TIFF.

- [PDFBitDepth](#)
Sets the bit depth of the image output by PDFDocumentToImage.
- [PDFCompression](#)
Sets the compression method used in the TIFF output by PDFDocumentToImage.
- [PDFConversionMethod](#)
Sets the conversion method used in the TIFF output by PDFDocumentToImage.
- [PDFDocumentToImage](#)
Create a TIFF image for each page in a PDF file.
- [PDFGrayscale](#)
Sets the output by PDFDocumentToImage to be grayscale.
- [PDFHorizontalResolution](#)
Sets the output horizontal resolution for PDFDocumentToImage.
- [PDFQuality](#)
Sets the conversion quality for PDFDocumentToImage.
- [PDFVerticalResolution](#)
Sets the output vertical resolution for PDFDocumentToImage.

Parent topic: [Convert actions](#)

PDFBitDepth

Sets the bit depth of the image output by PDFDocumentToImage.

Member of namespace

Convert

Syntax

```
bool PDFBitDepth (int p_iVal)
```

Parameters

p_iVal
Type: int

Parameters

One of the following bit depth values are allowed.

- 1 : black and white.
- 8 : 256 color.
- 24 : True color.

Returns

Always True.

Level

Page level.

Details

Sets the bit depth for the output TIFF. The number of bits determine the color capacity of an image. If this action is not called, a default value of 1 is used. This action must be called before PDFDocumentToImage.

Example:

```
PDFBitDepth(1)
PDFDocumentToImage()
```

Parent topic: [Pdf actions](#)

PDFCompression

Sets the compression method used in the TIFF output by PDFDocumentToImage.

Member of namespace

Convert

Syntax

```
bool PDFCompression (int p_iVal)
```

Parameters

p_iVal
Type: int

Parameters

One of the following compression values are allowed.

- 1 : No compression.
- 2 : CCITT modified Huffman RLE.
- 3: CCITT Group 3 fax.
- 4: CCITT Group 4 fax.
- 5: LZW Lempel-Ziv and Welch
- 7 : JPEG

Returns

Always True.

Level

Page level.

Details

Sets the compression of the output image from `PDFDocumentToImage`. If this action is not called, the default value of `CCITT4` will be used. Typically, input documents for recognition are black and white with fax compression.

Example:

```
PDFBitDepth(1)
PDFCompression(4)
PDFDocumentToImage()
```

Parent topic: [Pdf actions](#)

PDFConversionMethod

Sets the conversion method used in the TIFF output by `PDFDocumentToImage`.

Member of namespace

Convert

Syntax

```
bool PDFConversionMethod (int p_iVal)
```

Parameters

`p_iVal`
Type: int

Parameters

A numeric value of one of the following.

- 1 : Uses method 1 of converting PDF to TIFF.
- 2 : Uses method 2 of converting PDF to TIFF.

Returns

Always True.

Level

Page level.

Details

There are two methods that can be used to convert a PDF to TIFF. The first method is slightly faster. The second method gives better results. If this action is not called, `PDFDocumentToImage` uses the second method by default, which produces a more accurate output TIFF.

Example:

```
PDFBitDepth(1)
PDFConversionMethod(2)
PDFDocumentToImage()
```

Parent topic: [Pdf actions](#)

PDFDocumentToImage

Create a TIFF image for each page in a PDF file.

Member of namespace

Convert

Syntax

```
bool PDFDocumentToImage ()
```

Parameters

None.

Returns

True if the file is successfully converted to a TIFF document.

False if the current page is not a PDF or if there is a failure in the conversion.

If the number of input files/pages exceeds the maximum allowed or if there is a failure in the conversion, the batch is set to abort.

Level

Page level.

Details

If the current page is a PDF, the file will be converted to multiple TIFF files, one TIFF file for each page within the document.

Each new TIFF also has a new page created within the application environment which can be processed by subsequent rules. The original file name from which the page was extracted is stored in the *ParentImage* variable, for possible future reference within your application.

Example:

```
PDFBitDepth(1)
PDFDocumentToImage()
```

Restriction: This Pdf action does not support Unicode characters in the file name or file path. If your content uses Unicode characters in the file names or file paths, use the PdfFRE action instead.

Tip: When using non-native system fonts, use the PdfFRE class instead of Pdf.

Parent topic: [Pdf actions](#)

PDFGrayscale

Sets the output by PDFDocumentToImage to be grayscale.

Member of namespace

Convert

Syntax

```
bool PDFGrayscale (bool p_bVal)
```

Parameters

p_bVal
Type: bool

Parameters

A Boolean value to set grayscale mode.

True: The resulting TIFF is converted to grayscale.

False: The resulting TIFF is not converted to grayscale.

Returns

Always True.

Level

Page level.

Details

Sets the PDF conversion to TIFF so the output images are grayscale. This action must be called before PDFDocumentToImage. If this action is not called, a default value of False is used by PDFDocumentToImage.

Example:

```
PDFGrayscale (True)  
PDFDocumentToImage ()
```

Parent topic: [Pdf actions](#)

PDFHorizontalResolution

Sets the output horizontal resolution for PDFDocumentToImage.

Member of namespace

Convert

Syntax

```
bool PDFHorizontalResolution (int p_iVal)
```

Parameters

p_iVal
Type: int

Parameters

p_iVal : A positive numeric value for the horizontal resolution in dots per inch (DPI).

Returns

Always True.

Level

Page level.

Details

Sets the horizontal resolution for PDF conversion to TIFF. It is recommended that the horizontal and vertical resolutions be kept the same, creating an isotropic image. If this action is not called, the default resolution of 200 is used. This action must be called before PDFDocumentToImage.

Example:

```
PDFBitDepth(1)  
PDFHorizontalResolution(200)  
PDFVerticalResolution(200)  
PDFDocumentToImage()
```

Parent topic: [Pdf actions](#)

PDFQuality

Sets the conversion quality for PDFDocumentToImage.

Member of namespace

Convert

Syntax

```
bool PDFQuality (int p_iVal)
```

Parameters

p_iVal
Type: int

Parameters

p_iVal : A positive numeric value between 0 and 100 that determines the image quality. 100 is the highest quality.

Returns

Always True.

Level

Page level.

Details

This determines the quality of the output TIFF. Choosing a higher value will give you a better output TIFF, but will increase processing time. This action must be called before PDFDocumentToImage. If PDFQuality is not called, the default value of 100 is used.

Example:

```
PDFBitDepth(1)
PDFQuality(100)
PDFDocumentToImage()
```

Parent topic: [Pdf actions](#)

PDFVerticalResolution

Sets the output vertical resolution for PDFDocumentToImage.

Member of namespace

Convert

Syntax

```
bool PDFVerticalResolution (int p_iVal)
```

Parameters

p_iVal
Type: int

Parameters

p_iVal : A positive numeric value for the vertical resolution in dots per inch (DPI).

Returns

Always True.

Level

Page level.

Details

Sets the vertical resolution for PDF conversion to TIFF. It is recommended that the horizontal and vertical resolutions be kept the same, creating an isotropic image. If this action is not called, the default resolution of 200 is used. This action must be called before `PDFDocumentToImage`.

Example:

```
PDFBitDepth(1)
PDFHorizontalResolution(200)
PDFVerticalResolution(200)
PDFDocumentToImage()
```

Parent topic: [Pdf actions](#)

PdfFRE actions

Use the PdfFRE actions can use the Abbyy FineReader Engine to convert an image file from PDF format to TIFF for recognition processing by Datacap.

The PdfFRE actions can specify bit depth of the output image, as well as the compression algorithm and conversion method to use when you convert from PDF to TIFF.

- [PDFFREDocumentToImage](#)
Converts a PDF file to TIFF format and extracts any searchable text.
- [PDFFREReleaseEngine](#)
Releases resources used by the conversion engine.
- [PDF image compression types](#)
An image from a PDF document can be compressed in a number of different ways.

Parent topic: [Convert actions](#)

PDFFREDocumentToImage

Converts a PDF file to TIFF format and extracts any searchable text.

Member of namespace

Convert

Syntax

```
bool PDFFREDocumentToImage (string resolution, string compressionBW, string
compressionColor, string compressionGray, string extensionBW, string extensionColor,
string extensionGray, string convertMode, string useFastBinarization, string
jpegQuality)
```

Parameters

Smart parameters are supported.

resolution

The resolution of extracted images. The valid values are 50 – 3200 dots per inch.

compressionBW

The compression of extracted black and white pages in the source PDF. For more information about the valid types, see [PDF image compression types](#).

compressionColor

The compression of extracted color pages in the source PDF. For more information about the valid types, see [PDF image compression types](#).

compressionGray

The compression of extracted grayscale pages in the source PDF. For more information about the valid types, see [PDF image compression types](#).

extensionBW

The file extension of extracted black and white pages in the source PDF.

extensionColor

The file extension of extracted color pages in the source PDF.

extensionGray

The file extension to use for grayscale pages that are extracted from the source PDF.

convertMode

Used to set the conversion mode. Following are the possible values:

0	Preserve color.
1	Convert all images to black and white.

useFastBinarization

This setting is relevant when you set the convertMode to 1. It causes the conversion function to use a faster binarization algorithm to convert pages to black and white during extraction. Following are the possible values:

True	Use the faster binarization algorithm, which might result in lower quality images.
False	Do not use the faster binarization algorithm.

jpegQuality

The quality of the color images that are extracted with JPEG compression. The valid levels are 0 – 100. The higher the value, the higher the quality.

Returns

True if the file is successfully converted to a TIFF document. False if the current page is not a supported Image type or if failure in the conversion occurs. If the number of input files/pages exceeds the maximum that is allowed or if failure in the conversion occurs, the batch is set to abort.

Level

Page level.

Details

If the current page is a PDF, the file is converted to multiple TIFF files, one TIFF file for each page within the document.

If the PDF document contains searchable text, a CCO file that contains the positions and text is also created for each page in the document.

Note: Call the NormalizeCCO action, from the CCO2CCO action library, in a subsequent ruleset to ensure the integrity of the CCO file. It is also needed if the application is using the navigation and pattern match actions to

find recognized text on a page or perform pattern matching.

Each new TIFF also has a new page that is created within the application environment, which can be processed by subsequent rules. The original file name from which the page is extracted is stored in the *ParentImage* variable, for possible future reference within your application.

To prevent creating a CCO and to ignore searchable text within a PDF, enable *convPdfIgnoreContent* by setting the variable to "1" in the page DCO before you call *PDFDocumentToImage*. When *y_createLayout* is set to "1", then *convPdfIgnoreContent* is automatically enabled.

It is recommended to turn OFF the CCO file creation feature, in the case where it is unlikely that the application might process searchable PDF documents, or in the case where full page OCR is needed later in the workflow.

Creating a Layout File

This action runs recognition in addition to text extraction. Before you call the action, enable this capability by setting the DCO variable *y_createLayout* to "1". By default, this feature is turned off.

When this option is turned on, a layout xml file (for example *tm000001_layout.xml*) is created per image that is extracted.

Note: When *y_createLayout* is enabled, a CCO file is not created by default. Use the action *CreateCcoFromLayout* in the *SharedRecognitionTools* library on each newly created page to convert each layout XML to a CCO to allow other CCO actions to operate on the text.

The layout file groups text into blocks as a person would look at the document. Each block might have the default type of block or a specific type such as title or table. Locate actions are available in the *DocumentAnalytics* action library to navigate in the block structure such as *GoSiblingBlockNext*. Whereas the CCO file, that is produced by other actions, groups text into lines that span the width of the page.

The layout XML file also retains font and color attributes, which are saved in CSS format, for the text, which is used for extracting data and reconstructing the document in a new format.

To use the Locate actions and perform click 'n' key during verification, use the action *CreateCcoFromLayout* action in the *SharedRecognitionTools* action library. This action creates a CCO file for the page after the layout XML file is produced.

Following are the types of elements (in the "Block Type/XML Node" format) that might be present in the layout XML file:

- Block/Block
- Header/Header
- Footer/Footer
- Title/Title
- Heading1/H1
- Heading2/H2
- Heading3/H3
- Picture/Picture
- Barcode/Barcode
- Space/S
- Tab/Tab
- Table/Table
- Row/Row
- Cell/Cell
- Paragraph/Para

- Line/L
- Sentence/Sent
- Word/W
- Character/C

Text Extraction versus Text Recognition

By default, the text included in the layout XML is obtained from a combination of automatic recognition that is run on each page of the PDF and from searchable text that is embedded within the PDF. Any images that are embedded on the page have the text that is recognized by the engine.

If areas of the page contain both an image and searchable text that is associated with the image, the engine decides whether the engine must use the searchable text or recognize the text from the matching image. Because the engine performs recognition, the confidence of the text might vary even if the same searchable text is embedded in the PDF.

The variable `y_contentReuseMode` can be used to force the engine to use only the recognized text on the page or only to use the embedded text on the page. One reason why you might decide only to use the embedded text is to prevent recognition and produce high confidence results.

A drawback of using the only embedded text is that if the embedded text is wrong or incomplete, recognition is not performed to capture that missing data. The resultant layout XML that is created is incomplete compared to what the user sees when the user views the PDF. Do not use this setting if the source PDF file is of the image-on-text type because in this case, the text layer is not extracted. If a text line contains characters that are not included in the alphabet of the selected recognition languages, this text is not be written to the result. Mode 0 or 1 must be used.

These settings of `y_contentReuseMode` can be set on the DCO node that is being converted:

- `rrSet("0", "@X.y_contentReuseMode")` - The default auto mode that uses a combination of recognition and embedded text.
- `rrSet("1", "@X.y_contentReuseMode")` - Only recognition is used to create the layout XML.
- `rrSet("2", "@X.y_contentReuseMode")` - Only embedded text is used to create the layout XML.

For more information about configuring the recognition language, refer the OCR_A action, [Recognize](#).

Including PDF Annotations

By default, text annotations included in the source PDF file are not included in the output image. "Free Text" annotations in source PDF can be included in the output image by setting the page DCO variable `y_IncludeAnnotation` to "1". Other types of PDF annotations are not supported, such as popup and ink annotations. This feature does not cause the text of a "Sticky note" to be displayed on the image and a sticky note icon might display on the final image regardless of this setting.

Example

```
PDFFREDocumentToImage(300,32,32,32,".bw.tif", ".color.tif", ".gray.tif", 0, false,100)
```

Parent topic: [PdfFRE actions](#)

PDFFREReleaseEngine

Releases resources used by the conversion engine.

Member of namespace

Syntax

```
bool PDFFREReleaseEngine ()
```

Returns

Always True.

Level

Any level. Typically at the Batch close.

Details

Releases resources used by the conversion engine. This action should only be called after all the PDFs have been processed or the engine is loaded, unloaded, loaded etc. You need to place this action in a rule that is called at the batch close and it is to be called after all pages are processed.

Example:

```
PDFFREReleaseEngine(300,32,32,32, ".bw.tif", ".color.tif", ".gray.tif", 0,
false,100)
```

This example performs the conversion, then releases all resources used by the engine.

Parent topic: [PdfFRE actions](#)

PDF image compression types

An image from a PDF document can be compressed in a number of different ways.

Here are the possible values that you can pass to compression-related parameters:

1	BmpBwUncompressed
2	BmpGrayUncompressed
3	BmpColorUncompressed
4	DcxBwPackbits
5	DcxGrayPackbits
6	DcxColorPackbits
7	JpegGrayJfif
8	JpegColorJfif
9	PcxBwPackbits
10	PcxGrayPackbits
11	PcxColorPackbits
12	PngBwPng
13	PngGrayPng

14	PngColorPng
15	TiffBwUncompressed
16	TiffBwCcittGroup3
18	TiffBwCcittGroup4
19	TiffBwPackBits
20	TiffGrayUncompressed
21	TiffGrayPackBits
22	TiffGrayJpegJfif
23	TiffColorUncompressed
24	TiffColorPackBits
25	TiffColorJpegJfif
28	Jpeg2kGray
29	Jpeg2kColor
31	TiffBwLZW
32	TiffGrayLZW
33	TiffColorLZW
34	TiffBwZip
35	TiffGrayZip
36	TiffColorZip
43	JBIG2

Parent topic: [PdfFRE actions](#)

Rtf actions

Use the Rtf actions to convert an image file from RTF format to TIFF for recognition processing by Datacap.

The Rtf actions can specify the output resolution and compression algorithm that is used when you convert from RTF to TIFF.

- [RtfPrintQuality](#)
Adjusts the resolution of the image output by RtfToImage.
- [RtfTiffCompression](#)
Sets the compression used in the TIFF output by RtfToImage.
- [RtfToImage](#)
Converts a page with *.rtf file to a page or pages in TIFF format.
- [RtfToPdf](#)
Converts .rtf files to PDF document format.

Parent topic: [Convert actions](#)

RtfPrintQuality

Adjusts the resolution of the image output by RtfToImage.

Member of namespace

Convert

Syntax

```
bool RtfPrintQuality (int dpi)
```

Parameters

dpi

Type: int

A single positive numeric value for the dots per inch (dpi) of the output image.

Returns

False if the parameter is invalid. Otherwise, True.

Level

Page level.

Details

Sets the resolution of the output image for `RtfToImage`. If this action is not called, the default value of 200 dpi is used. Typically, input documents for recognition are 200 dpi.

Example:

```
RtfPrintQuality(200)  
RtfToImage()
```

Parent topic: [Rtf actions](#)

RtfTiffCompression

Sets the compression used in the TIFF output by `RtfToImage`.

Member of namespace

Convert

Syntax

```
bool RtfTiffCompression (string tiffCompression)
```

Parameters

tiffCompression

Type: string

A parameter of one of the following values to set the TIFF compression:

Parameters

- NONE : A color image with no compression.
- LZW : A color image using LZW compression.
- CCITT4 : A black and white image with fax CCITT4 compression.

Returns

Always True.

Level

Page level.

Details

Sets the compression of the output image from RtfToImage. If this action is not called, the default value of CCITT4 is used. Typically, input documents for recognition are black and white with fax compression.

Example:

```
RtfTiffCompression (CCITT4)  
RtfToImage ()
```

Parent topic: [Rtf actions](#)

RtfToImage

Converts a page with *.rtf file to a page or pages in TIFF format.

Member of namespace

Convert

Syntax

```
bool RtfToImage ()
```

Parameters

None.

Returns

True if the file is successfully converted to a TIFF document.

False if the current page is not a Rtf Document or if there is a failure in the conversion.

If the number of input files/pages exceeds the maximum allowed or if there is a failure in the conversion, the batch is set to abort.

Level

Page level.

Details

If the current page is a Rtf Document, the file will be converted to multiple TIFF files, one TIFF file for each page within the Document, based on the settings of the other Rtf actions that configure the conversion settings.

Each new TIFF also has a new page created within the application environment which can be processed by subsequent rules. The original file name from which the page was extracted is stored in the *ParentImage* variable, for possible future reference in your application.

If the configured output image format and compression only supports black and white, such as CCITT4, colored text is exported as black.

Example:

```
RtfPrintQuality(200)
RtfTiffCompression(CCITT4)
RtfToImage()
```

Parent topic: [Rtf actions](#)

RtfToPdf

Converts .rtf files to PDF document format.

Member of namespace

Convert

Syntax

```
bool RtfToPdf ()
```

Returns

True if the file conversion succeeds. Otherwise, the action returns False if the current page is not an RTF document or some other conversion failure occurs..

Level

Page level.

Details

Use this action to convert RTF files to a single PDF file. For information about setting the compliance standard for the produced PDF document, see [PDF compliance standards](#).

This action sets the batch to abort in the following circumstances:

- The number of input files or pages exceeds the allowed maximum.
- The conversion fails.

Example:

```
RtfToPdf
```

Parent topic: [Rtf actions](#)

Tiff actions

Use the Tiff action to convert a multi-page TIFF image file into multiple single page TIFF image files for recognition processing by Datacap.

The Tiff actions can split a multi-page TIFF image file into individual pages and set the compression method that is used by the SplitMultipageTiff action in the TIFF output.

- [SplitMultipageTiff](#)
Creates separate images for each page in a multipage Tiff file.
- [SplitTIFFCompression](#)
Sets the compression method used in the TIFF output by SplitMultipageTiff.

Parent topic: [Convert actions](#)

SplitMultipageTiff

Creates separate images for each page in a multipage Tiff file.

Member of namespace

Convert

Syntax

```
bool SplitMultipageTiff ()
```

Parameters

None.

Returns

True if the file is successfully converted to a TIFF document.

False if the current page is not a TIFF or if there is a failure in the conversion.

If the number of input files/pages exceeds the maximum allowed or if there is a failure in the conversion, the batch is set to abort.

Level

Page level.

Details

If the current page is a TIFF, the file is converted to multiple TIFF files, one TIFF file for each page within the document. If the input page was a single page TIFF file, a single page TIFF is still be output.

Each new TIFF also has a new page created within the application environment which can be processed by subsequent rules. The original file name from which the page was extracted is stored in the *ParentImage*

variable, for possible future reference in your application.

If the input TIFF has a bit depth of 1 (black and white), then the output compression is set to group 4 fax. If the input TIFF has a bit depth greater than 1, then the output compression is uncompressed.

Example:

```
SplitMultipageTiff()
```

Parent topic: [Tiff actions](#)

SplitTIFFCompression

Sets the compression method used in the TIFF output by `SplitMultipageTiff`.

Member of namespace

Convert

Syntax

```
bool SplitTIFFCompression (int compressionTypeColor, int compressionTypeBW)
```

Parameters

`compressionTypeColor`

Type: int

`compressionTypeBW`

Type: int

Parameters

One of the following compression values are allowed.

- 0 : Source Image compression.
- 1 : No compression.
- 2 : CCITT modified Huffman RLE. (BW only)
- 3 : CCITT Group 3 fax. (BW only)
- 4 : CCITT Group 4 fax. (BW only)
- 5 : LZW Lempel-Ziv and Welch

Returns

True if the compression setting was successful.

False if the compression setting used is not a supported type.

Level

Page level.

Details

Sets the compression of the output image from SplitMultipageTiff. If this action is not called, the default value of CCITT4 is used for black and white pages and No compression for color pages.

Example:

```
SplitTIFFCompression(4,1)
plitMultipageTiff()
```

Parent topic: [Tiff actions](#)

Txt actions

Use the Tiff actions to convert an image file from TXT format to TIFF for recognition processing by Datacap.

The Tiff actions can specify the output resolution and the compression algorithm that is used when you convert from Text to TIFF.

- [TxtFontName](#)
Adjusts the font name of the text in the image that is output by the TxtToImage action.
- [TxtFontSize](#)
Adjusts the font size of the text in the image that is output by the TxtToImage action.
- [TxtPrintQuality](#)
Adjusts the resolution of the image output by TxtToImage.
- [TxtTiffCompression](#)
Sets the compression used in the TIFF output by TxtToImage.
- [TxtToImage](#)
Converts a page with *.txt file to a page or pages in TIFF format.
- [TxtToPdf](#)
Converts .txt files to PDF document format.

Parent topic: [Convert actions](#)

TxtFontName

Adjusts the font name of the text in the image that is output by the TxtToImage action.

Member of namespace

Convert

Syntax

```
bool TxtFontName(string fontName)
```

Parameters

fontName
Type: string
The font name to use in the output image.

Returns

False if the parameter is invalid. Otherwise, True.

Level

Page level.

Details

Sets the font name for the text in the output image for the TxtToImage action. If this action is not called, the default value of Times New Roman is used.

Example:

```
TxtFontName(Courier New)
TxtToImage()
```

Parent topic: [Txt actions](#)

TxtFontSize

Adjusts the font size of the text in the image that is output by the TxtToImage action.

Member of namespace

Convert

Syntax

```
bool TxtFontSize(int fontSize)
```

Parameters

fontSize

Type: int

The size of the font to use in the output image.

Returns

False if the parameter is invalid. Otherwise, True.

Level

Page level.

Details

Sets the font size for the text in the output image for the TxtToImage action. If this action is not called, the default value of 10 is used.

Example:

```
TxtFontSize(12)
TxtToImage()
```

Parent topic: [Txt actions](#)

TxtPrintQuality

Adjusts the resolution of the image output by TxtToImage.

Member of namespace

Convert

Syntax

```
bool TxtPrintQuality (int dpi)
```

Parameters

dpi

Type: int

A single positive numeric value for the dots per inch (dpi) of the output image.

Returns

False if the parameter is invalid. Otherwise, True.

Level

Page level.

Details

Sets the resolution of the output image for TxtToImage. If this action is not called, the default value of 200 dpi is used. Typically, input documents for recognition are 200 dpi.

Example:

```
TxtPrintQuality(200)  
TxtToImage()
```

Parent topic: [Txt actions](#)

TxtTiffCompression

Sets the compression used in the TIFF output by TxtToImage.

Member of namespace

Convert

Syntax

```
bool TxtTiffCompression (string tiffCompression)
```

Parameters

tiffCompression

Type: string

A parameter of one of the following values to set the TIFF compression:

Parameters

- NONE : A color image with no compression.
- LZW : A color image using LZW compression.
- CCITT4 : A black and white image with fax CCITT4 compression.

Returns

Always True.

Level

Page level.

Details

Sets the compression of the output image from `TxtToImage`. If this action is not called, the default value of CCITT4 is used. Typically, input documents for recognition are black and white with fax compression.

Example:

```
TxtTiffCompression (CCITT4)
TxtToImage ()
```

Parent topic: [Txt actions](#)

TxtToImage

Converts a page with *.txt file to a page or pages in TIFF format.

Member of namespace

Convert

Syntax

```
bool TxtToImage ()
```

Parameters

None.

Returns

True if the file is successfully converted to a TIFF document.

False if the current page is not a Txt Document or if there is a failure in the conversion.

If the number of input files/pages exceeds the maximum allowed or if there is a failure in the conversion, the batch is set to abort.

Level

Page level.

Details

If the current page is a Txt Document, the file are converted to multiple TIFF files, one TIFF file for each page within the Document, based on the settings of the other Txt actions that configure the conversion settings.

Each new TIFF also has a new page created within the application environment which can be processed by subsequent rules. The original file name from which the page was extracted is stored in the *ParentImage* variable, for possible future reference within your application.

If the configured output image format and compression only supports black and white, such as CCITT4, colored text is exported as black.

Example:

```
TxtPrintQuality(200)
TxtTiffCompression(CCITT4)
TxtToImage()
```

Parent topic: [Txt actions](#)

TxtToPdf

Converts .txt files to PDF document format.

Member of namespace

Convert

Syntax

```
bool TxtToPdf()
```

Returns

True if the file conversion succeeds. Otherwise, the action returns False if the current page is not an TXT document or some other conversion failure occurs..

Level

Page level.

Details

Use this action to convert TXT files to a single PDF file. For information about setting the compliance standard for the produced PDF document, see [PDF compliance standards](#).

This action sets the batch to abort in the following circumstances:

- The number of input files or pages exceeds the allowed maximum.
- The conversion fails.

Example:

```
TxtToPdf
```

Parent topic: [Txt actions](#)

Word actions

Use the Word actions to convert an image file from Microsoft Word format to TIFF for recognition processing by Datacap.

The Word actions can specify the output resolution and the compression algorithm that is used when you convert from Word to TIFF.

- [WordDocumentToImage](#)
Converts a page with *.doc or *.docx file to a page or pages in TIFF format.
- [WordDocumentToPdf](#)
Converts .doc or.docx files to PDF document format.
- [WordMonochromeQuality](#)
Adjusts binarization settings used by WordDocumentToImage.
- [WordPrintQuality](#)
Adjusts the resolution of the image output by WordDocumentToImage.
- [WordTiffCompression](#)
Sets the compression used in the TIFF output by WordDocumentToImage.

Parent topic: [Convert actions](#)

WordDocumentToImage

Converts a page with *.doc or *.docx file to a page or pages in TIFF format.

Member of namespace

Convert

Syntax

```
bool WordDocumentToImage ()
```

Parameters

None.

Returns

True if the file is successfully converted to a TIFF document.

False if the current page is not a Word Document or if there is a failure in the conversion.

If the number of input files/pages exceeds the maximum allowed or if there is a failure in the conversion, the batch is set to abort.

Level

Page level.

Details

If the current page is a Word Document, the file is converted to multiple TIFF files, one TIFF file for each page within the Document, based on the settings of the other Word actions that configure the conversion settings.

Each new TIFF also has a new page created within the application environment which can be processed by subsequent rules. The original file name from which the page was extracted is stored in the *ParentImage* variable, for possible future reference in your application. If the configured output image format and compression only supports black and white, such as CCITT4, colored text is exported as black.

Example:

```
WordPrintQuality(200)
WordTiffCompression("CCITT4")
WordDocumentToImage()
```

Parent topic: [Word actions](#)

WordDocumentToPdf

Converts .doc or.docx files to PDF document format.

Member of namespace

Convert

Syntax

```
bool WordDocumentToPdf ()
```

Returns

True if the file conversion succeeds. Otherwise, the action returns False if the current page is not a Word document or some other conversion failure occurs..

Level

Page level.

Details

Use this action to convert Word documents to a single PDF file. For information about setting the compliance standard for the produced PDF document, see [PDF compliance standards](#).

This action sets the batch to abort in the following circumstances:

- The number of input files or pages exceeds the allowed maximum.
- The conversion fails.

Example:

```
WordDocumentToPdf
```

Parent topic: [Word actions](#)
»

WordMonochromeQuality

Adjusts binarization settings used by WordDocumentToImage.

Member of namespace

Convert

Syntax

```
bool WordMonochromeQuality (int method, float threshold)
```

Parameters

method

Sets the binarization method to either `Threshold` or `FloydSteinbergDithering` by using one of the following values:

- 0 (for `Threshold`)
- 1 (for `FloydSteinbergDithering`)

threshold

A floating value between 0 and 1 for rendering brightness. Lower values are darker.

Returns

Always True.

Level

Any level.

Details

Defaults compression to `CCITT4` and sets the binarization settings used for black and white output. If this action is not called, the default values of `Threshold` and 0.5 will be used for monochrome compression.

Example:

```
WordMonochromeQuality(0, 0.45)  
WordDocumentToImage()
```

Parent topic: [Word actions](#)
«

WordPrintQuality

Adjusts the resolution of the image output by WordDocumentToImage.

Member of namespace

Convert

Syntax

```
bool WordPrintQuality (int dpi)
```

Parameters

dpi
Type: int

Parameters

dpi : A single positive numeric value for the dots per inch (dpi) of the output image.

Returns

False if the parameter is invalid. Otherwise, True.

Level

Page level.

Details

Sets the resolution of the output image for WordDocumentToImage. If this action is not called, the default value of 200 dpi is used. Typically, input documents for recognition are 200 dpi.

Example:

```
WordPrintQuality(200)  
WordDocumentToImage()
```

Parent topic: [Word actions](#)

WordTiffCompression

Sets the compression used in the TIFF output by WordDocumentToImage.

Member of namespace

Convert

Syntax

```
bool WordTiffCompression (string tiffCompression)
```

Parameters

tiffCompression
Type: string

Parameters

tiffCompression is one of the following values to set the TIFF compression:

- NONE : A color image with no compression.
- LZW : A color image using LZW compression.
- CCITT4 : A black and white image with fax CCITT4 compression.

Returns

Always True.

Level

Page level.

Details

Sets the compression of the output image from WordDocumentToImage. If this action is not called, the default value of CCITT4 is used. Typically, input documents for recognition are black and white with fax compression.

Example:

```
WordTiffCompression (CCITT4)
WordDocumentToImage ()
```

Parent topic: [Word actions](#)

Zip actions

Use the Zip actions to extract the images files in a compressed ZIP archive and convert them to separate TIFF files for recognition by Datacap.

The Zip actions can overwrite the files that exist when you extract from ZIP archives and sets the Password value that is used to extract files from password protected archives.

- [ZipOverwrite](#)
Controls overwriting files when extracting from ZIP archives.
- [ZipPassword](#)
Sets the password for Password protected archives that will be used by ZipUnPack.
- [ZipUnPack](#)
Extracts each file within a compressed file archive into separate files.

Parent topic: [Convert actions](#)

ZipOverwrite

Controls overwriting files when extracting from ZIP archives.

Member of namespace

Convert

Syntax

```
bool ZipOverwrite (bool overwrt)
```

Parameters

overwrt
Type: bool

Parameters

A Boolean value that indicates if a subsequent ZipUnPack action should overwrite existing files.

True: Overwrite files that already exist.

False: Do not overwrite files that already exist.

Returns

Always True.

Level

Page level.

Details

This action must be called before ZipUnPack. If this action is not called, then ZipUnPack uses the default value of True.

Example:

```
ChkDCOStatus ("49")
ZipOverwrite ("True")
ZipUnPack ()
SetDCOStatus ("75")
```

Parent topic: [Zip actions](#)

ZipPassword

Sets the password for Password protected archives that will be used by ZipUnPack.

Member of namespace

Convert

Syntax

```
bool ZipPassword (string pwd)
```

Parameters

pwd
Type: string

Parameters

pwd : The password for the archive. Smart parameters are supported.

Returns

Always True.

Level

Page level.

Details

The password provided is used to extract password protected archives. This action must be called before ZipUnPack.

Example:

```
ChkDCOStatus ("49")
ZipOverwrite ("True")
ZipPassword ("MySafePassword")
ZipUnPack ()
SetDCOStatus ("75")
```

Parent topic: [Zip actions](#)

ZipUnPack

Extracts each file within a compressed file archive into separate files.

Member of namespace

Convert

Syntax

```
bool ZipUnPack ()
```

Parameters

None.

Returns

True, if the contents of the compressed file is successfully extracted.

False, if the current page is not a PDF or if there is a failure in the extraction.

If the number of input files/pages exceeds the maximum that is allowed or if there is a failure in the extraction, the batch is set to abort.

Level

Page level.

Details

If the current page is compressed, the files that are contained in the archive are placed into the current batch directory. Each new file also has a new page that is created within the application environment that can be processed by subsequent rules. The original file name from which the page was extracted is stored in the *ParentImage* variable, for future reference within your application.

Example:

```
ChkDCOStatus ("49")
ZipUnPack ()
SetDCOStatus ("75")
```

In this example, the action checks if the current page is the type, Other, and tries to extract the page. If the extraction is successful, then the current page was a valid compressed file. The files are extracted from the compressed file and put in the batch directory. The next action sets the page status of the compressed file to Deleted. The action does not delete it from the batch, but stops further processing of the compressed file.

Parent topic: [Zip actions](#)

DatacapBOX actions

Use the DatacapBOX actions to move files between your Datacap system and Box.com.

Important:

During IBM Datacap installation, the installer places DatacapBOX connector standard rulesets and action library on the target system, at this location: C:\Datacap\RRS\Box

You cannot use the *DatacapBOX* connector directly from this location. You must manually copy the entire contents of this folder to the “Rules” folder of the Datacap application that is interacting with Box.

For example, if “TravelDocs” is the Datacap application that interacts with DatacapBOX connector, copy all the files from C:\Datacap\RRS\Box folder to C:\Datacap\TravelDocs\dco_TravelDocs\Rules

The word *Box* as used here refers to your account on Box.com. To use the DatacapBOX actions, configure the Datacap system with your Box.com account information. For information about this configuration, see the section [Box Connector actions](#).

DatacapBOX actions are associated with the following objects:

- Export
- Import
- [Export object](#)
To export files from the Datacap system to Box, configure settings on the Export object and call the Upload action to initiate the export process.
- [Import object](#)
To import files from Box into the Datacap system, configure settings on the Import object and call the Download action to initiate the import process.

Parent topic: [Global actions](#)

Export object

To export files from the Datacap system to Box, configure settings on the Export object and call the Upload action to initiate the export process.

Important: Box limits the number of API calls per second, which means that export failures are more likely to occur as the number of export threads increases. To avoid export failures, do not configure more than eight threads for a Box export ruleset. Decrease the thread number as necessary if your export logs indicate export failures that are caused by the Box call rate limitation. For information about Export task logs, see [Task log files](#). Also, the optimal number of threads can vary depending on your hardware, image or batch sizes, and network performance. For general information about configuring threads in Rulerunner, see [Rulerunner thread configuration](#).

The export settings govern the behavior of the export process in the following ways:

Category	Comment	Relevant settings
DCO objects	You export one of the following types of DCO objects: <ul style="list-style-type: none"> • Pages • Document-level PDF files 	<ul style="list-style-type: none"> • ProcessChildren • DocumentsToPDF
Metadata	You optionally export the following items as Box file metadata: <ul style="list-style-type: none"> • DCO variables • Fields 	<ul style="list-style-type: none"> • AddParentDataToPageMetadata • DCOVarsAreMetadata • FieldsAreMetadata
Location	You optionally specify the Box target folder in which the export process places the exported files. By default, the target folder is your Box root folder.	<ul style="list-style-type: none"> • TargetFolder • CreateBatchSubfolder
Existing file versions	You optionally specify how the export process handles any existing versions of a Box file that has the same name as an exported file. By default, any existing file versions are ignored.	<ul style="list-style-type: none"> • FailIfExists • OverwriteExistingFiles • ReplaceMetadata

- [AddParentDataToPageMetadata](#)
A setting that determines whether the export process includes parent data in the Box metadata for exported files.
- [CreateBatchSubfolder](#)
A setting that determines whether the export process creates a subfolder for each exported batch. A subfolder has the same name as the batch and contains the exported files for the batch.
- [DCOVarsAreMetadata](#)
A setting that determines whether the export process includes DCO variable values in the Box metadata for exported files.
- [DocumentsToPDF](#)
A setting that determines the type of DCO objects that are exported to Box: either document-level PDF files or pages.

- [FailIfFileExists](#)
A setting that determines whether the export process fails if the target folder has one or more existing versions of a file to be exported.
- [FieldsAreMetadata](#)
A setting that determines whether the export process includes field values in the Box metadata for exported files.
- [OverwriteExistingFiles](#)
A setting that determines whether the export process overwrites any existing Box versions of a file to be exported. This setting has no effect if the FailIfFileExists action setting is set to True.
- [ProcessChildren](#)
A setting that determines whether the export process exports child objects of the current DCO object to Box.
- [ReplaceMetadata](#)
A setting that determines whether the export process replaces the Box metadata for a file when exporting a new version of the file to Box. One set of metadata is associated with all versions of the same Box file.
- [TargetFolder](#)
A setting that specifies the Box folder (if any) in which the export process places exported files.
- [Upload](#)
Exports files to Box from the current DCO object based on the settings of the Export object.

Parent topic: [DatacapBOX actions](#)

AddParentDataToPageMetadata

A setting that determines whether the export process includes parent data in the Box metadata for exported files.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

The term *parent* refers to the parent objects of a page such as the parent document object and the parent batch object. This setting has two possible values:

True	Parent data is included in the Box metadata.
False	Parent data is not included in the Box metadata.

The default value for this setting is False. Smart parameters are supported.

The following settings affect the type of parent data included in the file metadata:

- FieldsAreMetadata
- DCOVarsAreMetadata

Member of namespace

DatacapBOX

Syntax

AddParentDataToPageMetadata (bool NewValue)

Level

Any level.

Example

```
AddParentDataToPageMetadata()  
Upload()
```

Parent topic: [Export object](#)

Related reference:

[FieldsAreMetadata](#)

[DCOVarsAreMetadata](#)

CreateBatchSubfolder

A setting that determines whether the export process creates a subfolder for each exported batch. A subfolder has the same name as the batch and contains the exported files for the batch.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

This setting has two possible values:

True	Batch subfolders are created. The exported files are placed in the subfolders.
False	Batch subfolders are not created. The exported files are placed in the Box target folder.

The default value for this setting is False.

The TargetFolder setting specifies the Box target folder in which the batch subfolders are created.

Member of namespace

DatacapBOX

Syntax

```
CreateBatchSubfolder (bool NewValue)
```

Level

Any level.

Example

```
CreateBatchSubfolder (True)  
Upload()
```

Parent topic: [Export object](#)

Related reference:

[TargetFolder](#)

DCOVarsAreMetadata

A setting that determines whether the export process includes DCO variable values in the Box metadata for exported files.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

This setting has two possible values:

True	DCO variable values are included in the Box metadata.
False	DCO variable values are not included in the Box metadata.

The default value for this setting is False.

By default, the exported DCO variable values belong to the exported object only (a page or document-level PDF file). This default behavior can be overridden by the `AddParentDataToPageMetadata` setting.

Member of namespace

DatacapBOX

Syntax

```
DCOVarsAreMetadata (bool NewValue)
```

Level

Any level.

Example

```
DCOVarsAreMetadata (True)  
Upload()
```

Parent topic: [Export object](#)

Related reference:

[AddParentDataToPageMetadata](#)

DocumentsToPDF

A setting that determines the type of DCO objects that are exported to Box: either document-level PDF files or pages.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

This setting has two possible values:

True	The document-level PDF files of the current batch are exported to Box. (The export process does not create any PDF files; it assumes that they were previously created.)
False	The pages of the current batch are exported to Box.

The default value for this setting is False.

Member of namespace

Syntax

```
DocumentsToPDF(bool NewValue)
```

Level

Any level.

Example

```
DocumentsToPDF(True)
Upload()
```

Parent topic: [Export object](#)

FailIfExists

A setting that determines whether the export process fails if the target folder has one or more existing versions of a file to be exported.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

This setting has two possible values:

True	The export process fails if it encounters an existing Box version of a file to be exported. Such a failure means that the export process does not continue. The failure does not affect the files that were exported before the process failed; they remain exported.
False	The export process does not fail if it encounters an existing Box version of a file to be exported. (The <code>OverwriteExistingFiles</code> setting determines whether any existing file versions are overwritten.)

The default value for this setting is `False`.

Member of namespace

DatacapBOX

Syntax

```
FailIfExists(bool NewValue)
```

Level

Any level.

Example

```
FailIfExists(True)
Upload()
```

Parent topic: [Export object](#)

Related reference:

FieldsAreMetadata

A setting that determines whether the export process includes field values in the Box metadata for exported files.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

This setting has two possible values:

True	Field values are included in the Box metadata.
False	Field values are not included in the Box metadata.

The default value for this setting is False.

By default, the exported field values belong to the exported object only (a page or document-level PDF file). This default behavior can be overridden by the `AddParentDataToPageMetadata` setting.

Member of namespace

DatacapBOX

Syntax

```
FieldsAreMetadata (bool NewValue)
```

Level

Any level.

Example

```
FieldsAreMetadata (True)  
Upload ()
```

Parent topic: [Export object](#)

Related reference:

[AddParentDataToPageMetadata](#)

OverwriteExistingFiles

A setting that determines whether the export process overwrites any existing Box versions of a file to be exported. This setting has no effect if the `FailIfExists` action setting is set to True.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

This setting has two possible values:

True	The last existing Box file version in the target folder is deleted.
-------------	---

	For example, suppose that three versions of file abc exist in Box, and you are exporting the abc file. The export process deletes version 3 of the file in Box.
False	None of the existing Box file versions are deleted.

In both cases, a new version of the file is added to Box.

The default value for this setting is False.

Member of namespace

DatacapBOX

Syntax

```
OverwriteExistingFiles (bool NewValue)
```

Level

Any level.

Example

```
OverwriteExistingFiles (true)
Upload()
```

Parent topic: [Export object](#)

Related reference:
[FailIfFileExists](#)

ProcessChildren

A setting that determines whether the export process exports child objects of the current DCO object to Box.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

This setting has two possible values:

True	Child DCO objects are exported in addition to the current DCO object.
False	Child DCO objects are not exported. Only the current DCO object is exported.

The default value for this setting is True.

Member of namespace

DatacapBOX

Syntax

```
ProcessChildren (bool NewValue)
```

Level

Any level.

Example

```
ProcessChildren (True)  
Upload ()
```

Parent topic: [Export object](#)

ReplaceMetadata

A setting that determines whether the export process replaces the Box metadata for a file when exporting a new version of the file to Box. One set of metadata is associated with all versions of the same Box file.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

This setting has two possible values:

True	The metadata for the exported Box file includes the newly exported metadata only. The previous Box metadata for the file is discarded.
False	The metadata for the exported Box file includes the following sets of metadata merged together: <ul style="list-style-type: none">• The newly exported metadata• The previous Box metadata for the file before the export If the same field exists in both sets of metadata, the value from the newly exported metadata is used.

The default value for this setting is True.

Member of namespace

DatacapBOX

Syntax

```
ReplaceMetadata (bool NewValue)
```

Level

Any level.

Example

```
ReplaceMetadata (True)  
Upload ()
```

Parent topic: [Export object](#)

TargetFolder

A setting that specifies the Box folder (if any) in which the export process places exported files.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

You can specify the target folder as a path. If the specified target folder does not exist, the export process creates it.

The default value for this setting is an empty string, which means that the target folder is your root Box folder. Smart parameters are supported.

The CreateBatchSubfolder setting affects the way that files are placed in the target folder.

Member of namespace

DatacapBOX

Syntax

```
TargetFolder (bool NewValue)
```

Level

Any level.

Example

```
TargetFolder ("Datacap")  
Upload ()
```

Parent topic: [Export object](#)

Related reference:

[CreateBatchSubfolder](#)

Upload

Exports files to Box from the current DCO object based on the settings of the Export object.

Member of namespace

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

DatacapBOX

Syntax

```
Upload ()
```

Returns

True if the export process succeeds. Otherwise, the action returns False.

For information about the causes of a failed export process, see the rulerunner log in the current batch folder. For more information, see [Analyze the Rulerunner log](#).

The FailIfExists setting partly determines what constitutes a successful export.

Level

Batch, document, or page level.

Details

Example:

```
// DCO objects
ProcessChildren(True)

// Metadata
AddParentDataToPageMetadata(True)
DCOVarsAreMetadata(True)
FieldsAreMetadata(True)

// Location
TargetFolder("Datacap")
CreateBatchSubfolder(True)

// Existing file versions
OverwriteExistingFiles(True)
ReplaceMetadata(True)

// Export process
Upload()
```

Parent topic: [Export object](#)

Related reference:

[FailIfFileExists](#)

Import object

To import files from Box into the Datacap system, configure settings on the Import object and call the Download action to initiate the import process.

Important: If any concurrently running threads might be using the same Box source folder, your Box-related tasks cannot be run in a multi-threaded configuration. Instead, use single-threaded tasks only. For information about configuring threads in Rulerunner, see [Rulerunner thread configuration](#). For information about the Box source folder, see [SourceFolder](#).

The import settings govern the behavior of the import process in the following ways:

Category	Comment	Relevant settings
Files	You determine the files that are imported from Box by specifying the following items: <ul style="list-style-type: none">The Box source folder from which files are importedThe allowed file extensions of imported filesThe number of files to import	<ul style="list-style-type: none">SourceFolderLookForExtensionsImportLimit

Category	Comment	Relevant settings
Reimport prevention	You specify a backup folder to prevent files from being imported again (by a later import process).	<ul style="list-style-type: none"> BackupFolder
DCO objects	You import files from Box as one of the following types of objects: <ul style="list-style-type: none"> Document-level pages Batch-level pages 	<ul style="list-style-type: none"> ImportAsDocumentType

- [BackupFolder](#)
 A setting that specifies the Box backup folder (if any) to which the import process moves the files that were imported from the source folder. The purpose of this file relocation is to prevent files from being imported more than once. All versions of an imported Box file are moved to the backup folder.
- [Download](#)
 Imports files from Box into the current batch based on the settings of the Import object.
- [ImportAsDocumentType](#)
 A setting that specifies the type for the document that the import process creates (if any) to hold imported files as child pages.
- [ImportLimit](#)
 A setting that specifies the maximum number of files that the import process imports from Box.
- [LookforExtensions](#)
 A setting that specifies a comma-delimited list of file extensions that the import process uses as a criterion for determining which files to import from Box. A file must have one of the specified extensions to be imported.
- [SourceFolder](#)
 A setting that specifies the Box source folder from which the import process imports files. The name of the source folder is case-sensitive. Subfolders within the source folder are not traversed; files are imported from the source folder only.

Parent topic: [DatacapBOX actions](#)

BackupFolder

A setting that specifies the Box backup folder (if any) to which the import process moves the files that were imported from the source folder. The purpose of this file relocation is to prevent files from being imported more than once. All versions of an imported Box file are moved to the backup folder.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

You can specify the backup folder as a path. If the specified backup folder does not exist, the import process creates it.

The default value for this setting is an empty string, which means that the import process does not move imported files to a backup folder. Smart parameters are supported.

Member of namespace

DatacapBOX

Syntax

```
BackupFolder(string NewValue)
```

Level

Any level.

Example

```
BackupFolder("InDatacap")  
Download()
```

Parent topic: [Import object](#)

Download

Imports files from Box into the current batch based on the settings of the Import object.

Member of namespace

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

DatacapBOX

Syntax

```
Download()
```

Returns

True in the following circumstances:

- One or more files are found in the Box source folder (as specified by the SourceFolder setting)
- All files in the Box source folder are successfully imported

Otherwise, the action returns False.

For information about the causes of a failed import process, see the rulerunner log in the current batch folder. For more information, see [Analyze the Rulerunner log](#).

Level

Batch level.

Details

Use this action to import files into the current batch from Box based on the settings of the Import object. Box data is converted in the following manner:

- Box files become page objects
- Box metadata becomes page DCO variables

If no files are found in the Box source folder, the status of the current batch is set to Pending.

Example:

```
// Files
SourceFolder("ToDatacap")
LookForExtensions("tif,tiff,jpg,jpeg")

// Reimport prevention
BackupFolder("InDatacap")

// DCO objects
ImportAsDocumentType("MyDocType")

// Import process
Download()
```

Parent topic: [Import object](#)

Related reference:

[SourceFolder](#)

ImportAsDocumentType

A setting that specifies the type for the document that the import process creates (if any) to hold imported files as child pages.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

The default value for this setting is an empty string, which means that the import process does not create a document to hold imported pages. Instead, files are imported as batch-level pages. Smart parameters are supported.

Member of namespace

DatacapBOX

Syntax

```
ImportAsDocumentType(string NewValue)
```

Level

Any level.

Example

```
ImportAsDocumentType("MyDocType")
Download()
```

Parent topic: [Import object](#)

ImportLimit

A setting that specifies the maximum number of files that the import process imports from Box.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

A setting value of -1 or 0 indicates that no maximum limit applies. This setting is the default setting.

Smart parameters are supported.

Member of namespace

DatacapBOX

Syntax

```
ImportLimit(string NewValue)
```

Level

Any level.

Example

Code	Comment
<pre>ImportLimit("-1") Download()</pre>	No maximum file limit applies to the import process.
<pre>ImportLimit("5") Download()</pre>	The import process imports no more than five files.

Parent topic: [Import object](#)

LookforExtensions

A setting that specifies a comma-delimited list of file extensions that the import process uses as a criterion for determining which files to import from Box. A file must have one of the specified extensions to be imported.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

The default value for this setting is an empty string, which means that imported files can have any file extension.

Member of namespace

DatacapBOX

Syntax

```
LookForExtensions(string NewValue)
```

Level

Any level.

Example

```
LookForExtensions ("tif,tiff,jpg,jpeg")
Download()
```

Parent topic: [Import object](#)

SourceFolder

A setting that specifies the Box source folder from which the import process imports files. The name of the source folder is case-sensitive. Subfolders within the source folder are not traversed; files are imported from the source folder only.

Important: You must configure the DatacapBOX action before you use this option. For more information, see [DatacapBOX actions](#).

The import process fails in the following circumstances:

- No source folder is specified
- The specified source folder does not exist

Smart parameters are supported.

Member of namespace

DatacapBOX

Syntax

```
SourceFolder(string NewValue)
```

Level

Any level.

Example

```
SourceFolder ("ToDatacap")
Download()
```

Parent topic: [Import object](#)

Related reference:

[Download](#)

Dcclip actions

Use the Dcclip action to clip a portion of each page image and save it as a separate TIFF file.

The Dcclip action uses the recognition zone coordinates of the current field and clips the specified region of each page to a separate TIFF file.

- [dci_clipfield](#)
During processing, clips the field on the Image file (.tif) of every source page represented by the bound Field object of the Document Hierarchy and generates a separate Image file (.tif) displaying the clipped field's contents.

Parent topic: [Global actions](#)

dci_clipfield

During processing, clips the field on the Image file (.tif) of every source page represented by the bound Field object of the Document Hierarchy and generates a separate Image file (.tif) displaying the clipped field's contents.

Syntax

```
bool dci_clipfield (strParam)
```

Parameters

Two comma-separated string values:

1. The Page Type that is to be assigned to the Image file containing the clipped field and its value.
Remember: The new Image file is represented by a new page in the current Page file. The Page Type value you assign will be used to identify pages with clipped fields.
2. The Page Status to be assigned to pages with clipped images of the bound Field object of the Document Hierarchy.
Attention: Be sure that the status you assign conforms to Page Status specifications used throughout your application.

Returns

False if either parameter is invalid. Otherwise True.

If the dci_clipfield action cannot locate the target field on a source page, the action will not generate an Image file for the clipped field and will not add a corresponding page to the current Page file (.xml).

Level

Field level only.

Details

During processing, clips the field on the Image file (.tif) of every source page represented by the bound Field object of the Document Hierarchy and generates a separate Image file (.tif) displaying the clipped field's contents.

The action also adds a page representing the new Image file to the current Page file.

Attention: If the Image ID assigned to the Image file representing the source page has this format: tm000001.tif. The Image ID of a clipped field's Image file adds one underscore character and a two digit index and has this format: tm000001_01.tif

The second pair in the batch will have these ID's: tm000002.tif and tm000002_01.tif. This assumes that a source page has only one clipped field.

Example

```
dci_clipfield(OfficePens_Page, 0)
```

Parent topic: [Dcclip actions](#)

DCImageFix actions

Use the DCImageFix actions to clean up and enhance page images.

The image INI settings indicate which processes are run by the ImageEnhance action. You configure these INI settings on the Zones tab in Datacap Studio.

- [ImageEnhance](#)
Runs image processing by using pre-configured image enhancement settings, typically from the imagefix.ini file.
- [LoadSettings](#)
Loads the image enhancement settings that are used by the ImageEnhance action to run image processing.
- [LoadSettings_FingerprintID](#)
Loads the specific ImageFix Settings file (.ini) that corresponds to the Fingerprint ID of the current page.

Parent topic: [Global actions](#)

ImageEnhance

Runs image processing by using pre-configured image enhancement settings, typically from the imagefix.ini file.

Member of namespace

DCImageFix

Syntax

```
bool ImageEnhance (string BackupFileExtension)
```

Parameters

String: BackupFileExtension

Parameters

The file extension that the action is to assign to the backup of the original Image file. For example: tio.

The extension must be at least one character. If the leading period is provided, at least one character must follow it. Long name file extensions are allowed.

Returns

False If the parameter is not 3 or 4 alphanumeric characters, or if an exception is encountered while the image is being enhanced. Otherwise, True.

Level

Page or Field Level.

Details

Initiates image processing to run a pre-configured set of image enhancements.

Include this action after a LoadSettings or LoadSettings_FingerprintID action.

Example:

```
LoadSettings (C:\ParentDir\Invoice\Process\ImageFix.ini)
ImageEnhance ("tio")
```

In this example, the ImageFix settings that are specified in ImageFix.ini are applied to every page in the batch. An example of the copied file name is "TM000001.tio".

```
LoadSettings (C:\ParentDir\Invoice\Process\ImageFix.ini)
ImageEnhance ("tio.tif")
```

This example uses a longer file extension, which preserves the original file type. This extension makes it easier to view the original file without renaming. An example of the copied file name is "TM000001.tio.tif".

Parent topic: [DCImageFix actions](#)

LoadSettings

Loads the image enhancement settings that are used by the ImageEnhance action to run image processing.

Member of namespace

DCImageFix

Syntax

```
bool LoadSettings (string BackupFileExtension)
```

Parameters

String: BackupFileExtension The value of the path to the ImageFix Settings file (.ini).

Attention: The action can also use Smart Parameter syntax, such as the '@PATH(string)' method, to specify the path.

Returns

False if the ImageFix Settings file that you specify as a parameter is not found. Otherwise, True.

Level

All.

Details

This action loads the settings that the ImageFix action uses to process all images in the current batch. The action's parameter includes the file's name and complete path to its location in the application's Process directory. As an alternative, the parameter can use a smart parameter such as @Path to designate the value of the path to the same Process directory.

Example:

```
LoadSettings (C:\ParentDir\Invoice\Process\ImageFix.ini)
ImageEnhance (tio)
```

Example:

This example loads the settings file using the path denoted by the ScanFixSettings key that is listed in the Paths.ini file. If the key points to a relative path, it would be converted to the appropriate full path and then use that path to find the settings:

```
LoadSettings (@PATH (ScanFixSettings) )
ImageEnhance (tio)
```

Parent topic: [DCImageFix actions](#)

LoadSettings_FingerprintID

Loads the specific ImageFix Settings file (.ini) that corresponds to the Fingerprint ID of the current page.

Member of namespace

DCImageFix

Syntax

```
bool LoadSettings_FingerprintID (string FingerprintsFolderPath)
```

Parameters

String: FingerprintsFolderPath The value of the path to the fingerprint folder.

Attention: The action can also use Smart Parameter syntax, such as the '@PATH(string)' method to specify the path.

Returns

False if a fingerprint-specific Settings file does not exist. Otherwise, True.

Level

Page level only.

Details

The action searches the fingerprint folder of the application for a fingerprint-specific ImageFix Settings file. The settings for these files are assigned during the Image Enhancement phase of Fingerprint Definition, using the tools in Rule Manager's Image Processing Setup dialog. (Chapter 3 of the Rule Manager Reference shows you how to define a fingerprint-specific ImageFix Settings file.)

Important: The name of a fingerprint-specific ImageFix Settings file is limited to the Fingerprint ID with the .ini extension: 1044.ini, for example.

Example:

```
LoadSettings_FingerprintID()
ImageEnhance (tio)
```

Parent topic: [DCImageFix actions](#)

DCO actions

Use the DCO actions to set up and modify the runtime batch hierarchy (runtime DCO) information.

The DCO actions can create documents, set up the status and type properties of an object, check the status of an object, and create a page data file for an object.

- [ChkConfidence](#)
Checks the confidence of all field data on child pages against a minimum acceptable confidence value (Parameter 1). If any fields in a page contain Low Confidence data, assigns the Page Status that is specified in Parameter 2 to the page.
- [ChkDCOStatus](#)
Confirms that the status of the Document Hierarchy's current object is identical to the status entered as the parameter.
- [ChkDCOType](#)
Confirms that the Type property of the Document Hierarchy's current object is identical to the type entered as the parameter.
- [ChkLastDCOTypeEx](#)
Checks that the Type property of the Document Hierarchy's previous object is identical to the type entered as the parameter.
- [ChkIntegrity](#)
Checks that the integrity of the batch, as detailed in the Page file of the current task, meets the integrity requirements set within the Document Hierarchy (Setup DCO).
- [ChkLastDCOType](#)
Checks that the Type property of the Document Hierarchy's previous object is identical to the type entered as the parameter.
- [ClearAltText](#)
Clears character and confidence values from the Character Array position specified by the parameter.
- [ClearDCO](#)
Clears all objects of the Document Hierarchy which are children of the bound object, and their variables.
- [CopyPD2DD](#)
Assigns the value in a Page object's PD (Page Data) variable to the Document object's DD (Doc Data) variable.
- [CountPagesToDocumentVar](#)
Counts the number of pages in the document.
- [CreateDocuments](#)
Arranges the contents of a task's Page file (for example, Recognition.xml) into documents based on the Document Integrity rules (min, max and order) specified in your application's Document Hierarchy, and assembles documents from the pages in the batch.
- [CreateFields](#)
Creates the Data file for a page in a batch. The Data file for the first page in the batch, tm000001, as an example, is tm000001.xml.
- [DeleteFields](#)
Deletes all child fields and characters from calling object of the Document Hierarchy.
- [IsDocumentCountMoreThan](#)
Compares the number of documents in a batch with a parameter that you specify to let you manage the document size of your batches.
- [IsFirstDocumentInBatch](#)
Checks to see if this document is the first document in a batch.
- [JoinPreviousDocument](#)
Specifies the document type to join with the current document.
- [PropagateToAltText](#)
Copies the character and confidence values from the first index of the Character Array (index 0) to the

- index specified by the parameter.
- [RemoveDocumentStructure](#)
Removes the document hierarchy from the batch.
 - [SaveImageInformation](#)
Gets the image information and stores it into the DCO.
 - [SetDCOStatus](#)
Assigns a value to the Status property of the current object of the Document Hierarchy.
 - [SetDCOType](#)
Assigns a value to the Type property of the current object of the Document Hierarchy.
 - [SetDocStatus](#)
Assigns a status to the current document.
 - [SetDocumentType](#)
The action assigns a Document Type to the current Document object of the Document Hierarchy.
 - [SetFldConfidence](#)
For a specific field, sets the confidence for all characters in the field to the same value.
 - [SetPageFingerprintID](#)
Assigns a value to the FingerprintID property of the selected Page object of the Document Hierarchy.
 - [SetPageStatus](#)
The action assigns a page status to an object of the Document Hierarchy.
 - [SetPageTemplateID](#)
Assigns a value to the FingerprintID property of the selected Page object of the Document Hierarchy.
 - [SetPageType](#)
The action assigns a Page Type to the current Page object of the Document Hierarchy.

Parent topic: [Global actions](#)

ChkConfidence

Checks the confidence of all field data on child pages against a minimum acceptable confidence value (Parameter 1). If any fields in a page contain Low Confidence data, assigns the Page Status that is specified in Parameter 2 to the page.

Syntax

```
bool ChkConfidence (StrParam)
```

Parameters

Two or three comma-separated values:

1. The Numeric value of the minimum confidence required. This value is superseded on a field-by-field basis, if the field's *ReqConf* variable is set.
2. The Numeric Page Status code to assign to any page that has one or more fields with Low Confidence data: if a field's *ConfidenceString* property contains a value lower than the first parameter. Subfields, line items, and so on, are included. Typically, "1" (Problem) is the value of this parameter. If only two parameters are specified, only pages with *Status=0* are checked by this action.
3. If a third parameter is supplied, these parameters specify the list of Page Statuses to be checked.

Returns

True, if all fields in all source pages are High Confidence. False, if any field has Low Confidence data, or if the parameters are not Numeric.

Level

All levels. This action operates on the entire batch regardless of the level to which its rule is bound.

Details

Checks the confidence of all field data on selected pages, which are selected by Page Status, against a minimum acceptable confidence value (Parameter 1). If any fields contain Low Confidence data, the page is marked with the status specified as a parameter.

Optionally, checks only pages of the status that is specified as Parameter 3.

Example:

```
ChkConfidence (8,1)
```

Parent topic: [DCO actions](#)

Related reference:

[ChkDCOStatus](#)

ChkDCOStatus

Confirms that the status of the Document Hierarchy's current object is identical to the status entered as the parameter.

Syntax

```
bool ChkDCOStatus (StrParam)
```

Parameters

The Numeric value of the status that you are checking.

Returns

True, if the DCO status matches the parameter value. Otherwise, False.

Level

All levels.

Details

Confirms that the status of the Document Hierarchy's current object is identical to the status entered as the parameter.

Example:

```
ChkDCOStatus(0)  
returns True, if the current object has a status equal to 0, and False, if it  
does not.
```

```
ChkDCOStatus(48)  
returns True, if the current object has a status equal to 48, and False, if it  
does not.
```

Parent topic: [DCO actions](#)

Related reference:

[SetDCOStatus](#)

[ChkDCOType](#)

ChkDCOType

Confirms that the Type property of the Document Hierarchy's current object is identical to the type entered as the parameter.

Syntax

```
bool ChkDCOType (StrParam)
```

Parameters

The String value of the Type property of the object you're checking.

Returns

True if the value of the DCO Type matches the parameter. Otherwise, False.

Level

All levels.

Details

Confirms that the Type property of the Document Hierarchy's current object is identical to the type entered as the parameter.

Example:

```
ChkDCOType (Invoice)  
SetPageStatus (1)
```

Applied at the Page level, the action returns True if the current object is an Invoices Page object (using the Invoices application as an example), and False if it is not.

This action will confirm the current DCO Type matches an expected type and take additional subsequent actions that follow this action. In this case, if the current DCO Type is Invoice, the page status is set to 1.

Parent topic: [DCO actions](#)

Related reference:

[SetDCOType](#)

[ChkDCOStatus](#)

[ChkLastDCOType](#)

ChkLastDCOTypeEx

Checks that the Type property of the Document Hierarchy's previous object is identical to the type entered as the parameter.

Member of namespace

DCO

Syntax

```
bool ChkLastDCOTypeEx (string prevPageToMatch, string chkObjLevel, string ignoreTypes)
```

Parameters

- prevPageToMatch - The previous object's DCO Type to compare, case sensitive.
- chkObjLevel - (Optional) Objects to compare:
 - 0 - All
 - 1 - Documents only
 - 2 - Pages only
 - 3 - Fields only
- ignoreTypes - (Optional) Type or comma-delimited list of Types to skip evaluation. If the prior object's Type is listed within the parameter then it will be skipped and if present, objects prior to it will be checked.

Returns

True if the Type property of the Document Hierarchy's previous non-skipped object matches the page type attempting to be matched. Otherwise, False.

Level

All.

Details

Checks that the Type property of the Document Hierarchy's previous object is identical to the type entered as the parameter. This action allows you to test the last DCO Type encountered so you can take specific subsequent steps based on that type.

Example:

```
ChkLastDCOTypeEx (Separator, ,)
```

Parent topic: [DCO actions](#)

ChkIntegrity

Checks that the integrity of the batch, as detailed in the Page file of the current task, meets the integrity requirements set within the Document Hierarchy (Setup DCO).

Syntax

```
bool ChkIntegrity ()
```

Parameters

None.

Returns

Returns True if no integrity problems are found. Otherwise, False.

Level

Batch and Document levels.

Details

Checks that the integrity of the batch, as detailed in the Page file of the current task, meets the integrity requirements set within the Document Hierarchy (Setup DCO).

Integrity refers to the correct types and numbers of pages within each document in the batch and the correct order of the pages in each document.

Example:

```
ChkIntegrity()
```

Parent topic: [DCO actions](#)

ChkLastDCOType

Checks that the Type property of the Document Hierarchy's previous object is identical to the type entered as the parameter.

Syntax

```
bool ChkLastDCOType (StrParam)
```

Parameters

The previous object's DCO Type to compare.

Returns:

True, if the Type property of the Document Hierarchy's previous object matches the parameter. Otherwise, False.

Level

All.

Details

Checks that the Type property of the Document Hierarchy's previous object is identical to the type entered as the parameter. You can use this action to test the last DCO Type that is encountered so you can take specific subsequent steps that are based on that type.

The example is applied at the Page level and checks to see whether the previous Page object's type matches the parameter (Separator).

Example:

```
ChkLastDCOType (Separator)
SetPageType (Invoice)
SetPageStatus (1)
```

Applied at the Page level, this sequence checks to see whether the previous Page object was a Separator page. If so, the type of the current page is set to Invoice, and its status is set to "1". If the previous document type was not a separator, the subsequent actions do not execute.

Parent topic: [DCO actions](#)

Related reference:

[SetDCOType](#)

[ChkDCOType](#)

ClearAltText

Clears character and confidence values from the Character Array position specified by the parameter.

Syntax

```
bool ClearAltText (strParam)
```

Parameters

The index in the Character Array where you want to clear the character and confidence values. 0 is the first index, followed by 1, etc.

Returns

Always True.

Level

Field level.

Details

Clears character and confidence values from the Character Array position specified by the parameter. When cleared, the confidence values are set to 10 (high confidence). (A field in the Data file can hold more than one representation of the field's value. Values other than the current, visible value are accessed via an index number. The current value is at index 0, the next value is at index 1. Each additional value also has corresponding character confidences.)

Note: Most actions only work with characters and confidence values located in the first position of the Character Array (position 0). The Cleartext action is used with User Application Web's Advanced Index task, for "Double Blind" data entry.

Example:

```
PropagateToAltText (1)
ClearAltText (0)
```

In this example, all characters at the first index (0) of the Character Array will be copied to the second index (1). The second action will then clear character and confidence values from the first index in the Character Array.

Parent topic: [DCO actions](#)

Related reference:

[PropagateToAltText](#)

ClearDCO

Clears all objects of the Document Hierarchy which are children of the bound object, and their variables.

Syntax

```
bool ClearDCO ()
```

Parameters

None.

Returns

True if all child objects and their variables are removed, otherwise False.

Level

All.

Details

Removes all DCO children and variable references from the bound object.

Example:

```
CreateFields ()  
ClearDCO ()
```

Applied at the Page level, the example will first add fields to the page and then remove them.

Parent topic: [DCO actions](#)

CopyPD2DD

Assigns the value in a Page object's PD (Page Data) variable to the Document object's DD (Doc Data) variable.

Syntax

```
bool CopyPD2DD ()
```

Parameters

None.

Returns

False if the action is not at the Document level, or if the PD variable at page level has no value. Otherwise, True.

Level

Document level.

Details

Assigns the value in a Page object's PD (Page Data) variable to the Document object's DD (Doc Data) variable.

Example:

```
CopyPD2DD ()
```

Parent topic: [DCO actions](#)

CountPagesToDocumentVar

Counts the number of pages in the document.

Syntax

```
bool CountPagesToDocumentVar (StrParam)
```

Parameters

None

Returns

Always True.

Level

Document level.

Details

This action counts the number of page objects in a document and writes the result in a document variable.

Example:

```
CountPagesToDocumentVar ("MyVarName")
```

Parent topic: [DCO actions](#)

CreateDocuments

Arranges the contents of a task's Page file (for example, Recognition.xml) into documents based on the Document Integrity rules (min, max and order) specified in your application's Document Hierarchy, and assembles documents from the pages in the batch.

Syntax

```
bool CreateDocuments ()
```

Parameters

None.

Returns

True if successful. Otherwise, False.

Level

Batch level only.

Details

Arranges the contents of a task's Page file (for example, Recognition.xml) into documents based on the Document Integrity rules for min, max, and order properties specified in your application's Document Hierarchy, and assembles documents from the pages in the batch.

Batches containing existing document structures will cause this action to return False, with no affect to the existing document structure.

Note: During document creation, temporary IDs are assigned (with a different format than real Document IDs), and if the action fails, these temporary IDs remain.

Important: This action is applied at the Batch level, and generally in its own Ruleset (in a CreateDocs ruleset, for example.)

Example:

```
CreateDocuments ()
```

Parent topic: [DCO actions](#)

CreateFields

Creates the Data file for a page in a batch. The Data file for the first page in the batch, tm000001, as an example, is tm000001.xml.

Syntax

```
bool CreateFields ()
```

Parameters

None.

Returns

True if successful. Otherwise, False.

Level

Page level.

Details

Creates the Data file for a page in a batch. The Data file for the first page in the batch, tm000001, as an example, is tm000001.xml.

This Data file lists all fields for the current page based on the fields listed in the setup Document Hierarchy. Each field has an ID (for an Invoices page, for example, the Date field's ID is Date), and three properties with default values: TYPE, Position, and Status.

Later, actions of various kinds (Locate, Zone, Validation, DCO, etc.) can assign other values to these properties. These actions can also add properties (variables) and values to the Data file, or remove properties and values.

Example:

```
AnalyzeImage ()
RotateImageRecognizePageOCR_S ()
SetSearchArea (0.5)
SetProblemValue (0.3)
SetTemplateDir (\ParentDirectory\Invoice\Template)
FindFingerprint (True)
CreateFields ()
```

This Invoices application sequence sets up the current page for processing, recognizes the words on the page, associates the page with a fingerprint, and finally creates a Data file with blank fields for the page.

Parent topic: [DCO actions](#)

DeleteFields

Deletes all child fields and characters from calling object of the Document Hierarchy.

Syntax

```
bool DeleteFields ()
```

Parameters

None.

Returns

True if successful, otherwise False.

Level

All.

Details

Deletes all fields and characters that are children of the bound object of the Document Hierarchy. This action will also remove the Data file (.xml) from the batch if called from a Page object.

Example:

```
DeleteFields()
```

Parent topic: [DCO actions](#)

IsDocumentCountMoreThan

Compares the number of documents in a batch with a parameter that you specify to let you manage the document size of your batches.

Syntax

```
bool IsDocumentCountMoreThan (string count , string returnTrueIfMore)
```

Parameters

`returnTrueIfMore`: determines whether the action returns True or False based on the results of the document count comparison.

Smart parameters are supported.

Returns

When set to True, returns True if the document count exceeds the value of the specified parameter. Otherwise, False.

When set to False, returns False if the document count exceeds the value of the specified parameter. Otherwise, True.

Level

All levels.

Details

This action compares the current batch document with the parameter that is provided and returns True or False depending on whether the count is exceeded by the provided parameter or not.

Example:

```
IsDocumentCountMoreThan(1, true)
```

This example returns True if there is more than 1 document in the batch

```
sDocumentCountMoreThan(1, false)
```

This example returns False if there is more than 1 document in the batch

Parent topic: [DCO actions](#)

IsFirstDocumentInBatch

Checks to see if this document is the first document in a batch.

Syntax

```
bool IsFirstDocumentInBatch()
```

Parameters

None

Returns

True if the action is on the first document, or an object in the first document. Otherwise False.

Level

Document level.

Details

This action checks to see if this action is on an object in the first document in a batch.

Parent topic: [DCO actions](#)

JoinPreviousDocument

Specifies the document type to join with the current document.

Syntax

```
bool JoinPreviousDocument (string DocTypeToCombine)
```

Parameters

None

Returns

Always True.

Level

Document, Page, or Field level.

Details

This action copies the pages of the previously named document into the front of the current document.

Example:

```
JoinPreviousDocument ("SeparatorDoc")
```

Parent topic: [DCO actions](#)

PropagateToAltText

Copies the character and confidence values from the first index of the Character Array (index 0) to the index specified by the parameter.

Syntax

```
bool PropagateToAltText (strParam)
```

Parameters

The index of the Character Array where you want to copy the character and confidence values. 0 is the first index, followed by 1, etc.

Returns

Always True.

Level

Field level.

Details

Copies the character and confidence values from the first index of the Character Array (index 0) to the index specified by the parameter. (The character "node" of a page's Data file is an array that can hold many recognized character values, and their corresponding confidence values.)

Note: Rules will only work with characters and confidence values located in the first position of the Character Array (index 0).

The PropagateToAltText action is used with the User Application Web's Advanced Index task, for "Double Blind" data entry.

Example:

```
PropagateToAltText (1)  
ClearAltText (0)
```

All characters in the first index of the Character Array will be copied to the second index. Then, the second action will clear the character and confidence values from the first index.

Parent topic: [DCO actions](#)

Related reference:

[ClearAltText](#)

RemoveDocumentStructure

Removes the document hierarchy from the batch.

Syntax

```
bool RemoveDocumentStructure ()
```

Parameters

None

Returns

Always True.

Level

Batch level.

Details

This action flattens the document and page hierarchy from the batch. If the batch consists of multiple documents, each with a set of pages, the document level is removed and all of the pages become a flat structure. Once complete, there is no distinction of sets of pages within a document.

Example:

```
RemoveDocumentStructure()
```

Parent topic: [DCO actions](#)

SaveImageInformation

Gets the image information and stores it into the DCO.

Syntax

```
bool SaveImageInformation(string ImageWidth, string ImageHeight, string ImageDPIX,  
string ImageDPIY, string PhysicalImageWidth, string PhysicalImageHeight, string  
BitDepth)
```

Parameters

string ImageWidth

string ImageHeight

string ImageDPIX

string ImageDPIY

string PhysicalImageWidth

string PhysicalImageHeight

string BitDepth

Parameters

ImageWidth: The input parameter is the name that user must specify to store the image width in pixels. If not specified, the DCO variable is not created for width.

ImageHeight: The input parameter is the name that user must specify to store the image height in pixels. If not specified, the DCO variable is not created for height.

ImageDPIX: The input parameter is the name that user must specify to store the DPIX in dots per inch. If not specified, the DCO variable is not created for DPIX.

ImageDPIY: The input parameter is the name that user must specify to store the DPIY in dots per inch. If not specified, the DCO variable is not created for DPIY.

PhysicalImageWidth: The input parameter is the name that user must specify to store the physical width in pixels. If not specified, the DCO variable is not created for physical width.

PhysicalImageHeight: The input parameter is the name that user must specify to store the physical height in pixels. If not specified, the DCO variable is not created for physical height.

BitDepth: The input parameter is the name that user must specify to store the depth in bit. If not specified, the DCO object is not created for Bit Depth.

Returns

True if the operation is successful, else False if any error occurred.

Level

Page level only, and the page must refer to a valid single page image file.

Details

This action gets the image information Height, Width, DPIX, DPIY, Physical Height, Physical Width, Bit Depth into respective parameters that are specified by user and stores it into the Document heirarchy.

Example 1:

```
SaveImageInformation("@P.Width", "@P.Height", "@P.DPIX", "@P.DPIY", "@P.PhysicalWidth", "@P.PhysicalHeight", "@P.BitDepth") RemoveDocumentStructure()
```

This example stores image information into variables that are specified by user at the page level.

Width = "2479"

Height = "3508"

DPIX = "300"

DPIY = "300"

PhysicalWidth = "743700"

PhysicalHeight = "1052400"

BitDepth = "1"

Example 2:

```
SaveImageInformation(" ", " ", " ", "@P.DPIX", "@P.DPIY", "@P.PhysicalWidth", "@P.PhysicalHeight", "@P.BitDepth")
```

This example stores image information DPIx, DPIy, PhysicalWidth, PhysicalHeight, BitDepth into respective variables that are specified at the page level.

```
DPIx = "300"
```

```
DPIy = "300"
```

```
PhysicalWidth = "743700"
```

```
PhysicalHeight = "1052400"
```

```
BitDepth = "1"
```

Example 3:

```
SaveImageInformation("@P\Field1.Width", "@P\Field1.Height", "@P\Field1.DPIx", "@P\Field1.DPIy", "@P\Field1.PhysicalWidth", "@P\Field1.PhysicalHeight", "@P\Field1.BitDepth")
```

This example stores image information into respective parameters that are specified by user at the field level.

```
Width = "2479"
```

```
Height = "3508"
```

```
DPIx = "300"
```

```
DPIy = "300"
```

```
PhysicalWidth = "743700"
```

```
PhysicalHeight = "1052400"
```

```
BitDepth = "1"
```

Parent topic: [DCO actions](#)

SetDCOStatus

Assigns a value to the Status property of the current object of the Document Hierarchy.

Syntax

```
bool SetDCOStatus (StrParam)
```

Parameters

An Integer representing the new status.

Returns

Always True.

Level

All.

Details

Assigns a value to the Status property of the current object of the Document Hierarchy.

Example:

```
ChkDCOType (Invoice)  
SetDCOStatus (1)
```

This sequence checks to see if the current object of the Document Hierarchy is a Page object - in this example, an Invoices page. If so, the value of the Page object's Status property is set to "1".

Parent topic: [DCO actions](#)

Related reference:

[SetDCOType](#)

[SetPageType](#)

SetDCOType

Assigns a value to the Type property of the current object of the Document Hierarchy.

Syntax

```
bool SetDCOType (StrParam)
```

Parameters

A String value you're assigning to the current object's Type property.

Returns

Always True.

Level

All.

Details

Assigns a value to the Type property of the current object of the Document Hierarchy.

Example:

```
ChkLastDCOType (Separator)  
SetDCOType (Invoice)
```

This sequence checks to see if the previous object of the Invoices application's Document Hierarchy was a Page object - in this case, a Separator page. If so, it sets Invoice as the Type property of the current object.

Parent topic: [DCO actions](#)

Related reference:

[SetDCOStatus](#)

[SetPageStatus](#)

[SetDocumentType](#)

SetDocStatus

Assigns a status to the current document.

Syntax

```
bool SetDocStatus (StrParam)
```

Parameters

String value representing the status to be assigned to the current document.

Typically:

- "0" = Complete.
- "1" = Incomplete.

Returns

False if the ruleset is not bound to a Document object, or the current object is not a document. Otherwise, True.

Level

Document level.

Details

Assigns a status to the current document.

Example:

```
SetDocStatus (DocOK)
```

Parent topic: [DCO actions](#)

SetDocumentType

The action assigns a Document Type to the current Document object of the Document Hierarchy.

Syntax

```
bool SetDocumentType (StrParam)
```

Parameters

The value you want to assign as the Document object's Type property.

You can also designate a field in a Page object's Data file, and use its text value to set the Document Type. Simply enter the name of a valid Field object and surround it with single quotes. For example: 'Number'.

Returns

False if there are no Document objects in the Data file, or if the parameter is invalid. Otherwise, True.

Level

Document, Page, and Field levels.

Details

Similar to the SetDCOType action but works at the Document, Page or Field level.

The action assigns the Document Type you enter as a parameter to the current Document object of the Document Hierarchy. You can also use a Field object's value to set the Document Type. (Refer to the Parameter section.)

Example:

```
SetDocumentType ('Number') \  
or  
SetDocumentType (Invoice_Document)
```

Parent topic: [DCO actions](#)

Related reference:

[SetDCOType](#)

[SetPageType](#)

SetFldConfidence

For a specific field, sets the confidence for all characters in the field to the same value.

Syntax

```
bool SetFldConfidence (StrParam)
```

Parameters

A comma-separated value that consists of:

1. The field name, or a Smart Parameter that designates the field.
2. The confidence value (1-10) to be assigned to the field's characters.

Returns

Always True. If an input parameter is invalid, no confidence levels are changed and a message is logged.

Level

Field level.

Details

This action unconditionally sets the confidence values for every character within a field to a specific level. This action can help change confidence levels when the preceding actions are all successful.

For example, a successful set of calculations that involves one or more fields might indicate that the confidence level of those fields' values ought to be high, regardless of the confidence set by the recognition engine. After

the success of the calculation, you can call `SetFldConfidence` to unconditionally reset the confidence values of the fields.

The first parameter is the name of the field that has its confidence level set. The second parameter is the wanted confidence level, between 1 and 10. If the second parameter is not passed in, 10 is used as the default.

Note: This function supports Smart parameters for the field name. See the following examples.

Example:

Example 1:

To set all characters in the field "GrossSalary" to a confidence of 9.
`SetFldConfidence(@P\GrossSalary,10)`

Example 2:

To set all characters in the field "AdjustedPay" to a confidence of 1.
`SetFldConfidence(@P\AdjustedPay,1)`

Example 3:

In the context of a test run before setting the confidence.

```
Calculate("'1TotalWages' + '2TaxableInterest' + '3Unemployment' =  
'4AdjustedGross'")
```

If this calculation works, the application can assume that all of the characters are read correctly and `SetFldConfidence` can adjust the fields to high confidence.

Note, you might also want to add a check that the values are all non-zero to eliminate a bad read.

```
SetFldConfidence("@P\1totalWages,10")  
SetFldConfidence("@P\2TaxableInterest,10")  
SetFldConfidence("@P\3Unemployment,10")  
SetFldConfidence("@P\4AdjustedGross,10")
```

Parent topic: [DCO actions](#)

SetPageFingerprintID

Assigns a value to the `FingerprintID` property of the selected Page object of the Document Hierarchy.

Syntax

```
bool SetPageFingerprintID (StrParam)
```

Parameters

The String value of the Fingerprint ID.

Returns

True if the rule is applied at the Page level. Otherwise, False.

Level

Page level.

Details

Assigns a value to the FingerprintID property of the selected Page object of the Document Hierarchy. Important: The SetPageFingerprintID action will create a FingerprintID property of the current Page object if it does not already exist.

Example:

```
WordFind (MQSW)  
SetPageFingerprintID (1010)
```

In this sequence, if the WordFind action locates "MQSW" on the current page, the SetPageFingerprintID action assigns "1010" as the page's Fingerprint ID. This links the page to a fingerprint with a Fingerprint ID of "1010".

Parent topic: [DCO actions](#)

SetPageStatus

The action assigns a page status to an object of the Document Hierarchy.

Syntax

```
bool SetPageStatus (StrParam)
```

Parameters

Numeric value that represents the status.

The Invoices application (as an example) employs three default Page Statuses:

- 49 = ScanOK
- 1 = Incomplete/Not validated
- 0 = Complete

You can define your own statuses by using the Filter tab of a task's Task Settings dialog.

Returns

Always True.

Level

Page or field level.

Details

The action assigns the page status that you enter to the page object of the Document Hierarchy. The current object can be the page or field. If the current object is a field object, it sets the page status for its parent page object.

Example:

A scan task might assign Other as the Page Type and 49 as the default Page Status for every successfully scanned image in the batch.

The following sequence is an example of a rule that converts Other pages to Invoices pages, and assigns a Page Status to each:

```
SetPageType (Invoice)  
SetPageStatus (1)
```

This combination establishes the page as an Invoices page, and gives it a status of 1. This means that the page is not validated and must be processed by a task that applies Validation rules (a Recognition or Verification task, for example).

Parent topic: [DCO actions](#)

Related reference:

[SetPageType](#)

[SetDCOStatus](#)

[SetDCOType](#)

[SetDocStatus](#)

SetPageTemplateID

Assigns a value to the FingerprintID property of the selected Page object of the Document Hierarchy.

Syntax

```
bool SetPageTemplateID (strParam)
```

Parameters

The String value of the Fingerprint ID.

Returns

True if the rule is applied at the Page level. Otherwise, False.

Level

Page level.

Details

The SetPageTemplateID action creates a FingerprintID property of the current Page object if one does not already exist.

Example:

```
WordFind (MQSW)  
SetPageTemplateID (1010)
```

In this sequence, if the WordFind action locates "MQSW" on the current page, the SetPageTemplateID action assigns "1010" as the page's Fingerprint ID. This links the page to a fingerprint with a Fingerprint ID of "1010".

Parent topic: [DCO actions](#)

SetPageType

The action assigns a Page Type to the current Page object of the Document Hierarchy.

Syntax

```
bool SetPageType (StrParam)
```

Parameters

A String value that represents the Page Type.

You can also designate a field in a Page object's Data file, and use its text value to set the Page Type. Enter the name of a valid Field object and surround it with single quotation marks. For example: 'PageCode'.

Returns

False, if there are no Page objects in the Page file, or if the parameter is invalid. Otherwise, True.

Level

Page and Field levels.

Details

Similar to the SetDCOType action, but works at the Page or Field level.

The action assigns the Page Type you enter as a parameter to the current Page object of the Document Hierarchy. You can also use a Field object's value to set the Page Type. (See the Parameter section.)

Example:

The application's scan task typically assigns Other as the Page Type and 49 as the default Page Status for every successfully scanned image in the batch.

The following sequence is an example of a rule that converts Other pages to Invoices pages, and assigns a Page Status to each:

```
SetPageType (Invoice)  
SetPageStatus (1)
```

This combination sets the page as an Invoices page, and gives it a status of 1. This means that the page is not validated and must be processed by a task that applies Validation rules (a Rulerunner Task, for example).

Parent topic: [DCO actions](#)

Related reference:

[SetPageStatus](#)

[SetDCOStatus](#)

[SetDCOType](#)

dcpdf actions

Use the dcpdf actions to convert PDF files to TIFF at the start of the workflow. You can also convert the TIFF files in a document into a PDF file.

The dcpdf actions can specify properties for PDF documents, set bits per pixel counts for images in PDF documents, and specify the compression method to use to convert PDF documents to TIFF.

- [dcpdf_CreateTiffFromPDF](#)
Converts each page of a PDF file into a TIFF file.
- [dcpdf_CreateTiffFromPDF_CreateDocs](#)
Converts each page of a PDF file into a TIFF file and creates a runtime hierarchy.
- [dcpdf_MakePDFDoc](#)
Creates a PDF document that contains one or more pages of the current document.
- [dcpdf_MaxSizeToReconvert](#)
Causes the dcpdf_CreateTiffFromPDF action to use a different conversion algorithm if the file that results from the default algorithm exceeds the specified size.
- [dcpdf_SetApplication](#)
Specifies the Application ID property for PDF documents that are generated by the dcpdf_MakePDFDoc action.
- [dcpdf_SetAuthor](#)
Specifies the Author property for PDF documents that are generated by the dcpdf_MakePDFDoc action.
- [dcpdf_SetImageBitcount](#)
Sets the bit count, bits per pixel, for images in the PDF document that is generated by the dcpdf_MakePDFDoc action.
- [dcpdf_SetImageCompression](#)
Specifies the compression method to use when you convert a PDF file to TIFF.
- [dcpdf_SetImageGrayscale](#)
Specifies how gray areas of a grayscale image are handled when you convert a PDF file to TIFF.
- [dcpdf_SetImageQuality](#)
Specifies the image quality to use when you convert a PDF file to TIFF.
- [dcpdf_SetImageResolution](#)
Specifies the output resolution to use when you convert a PDF file to TIFF.
- [dcpdf_SetKeywords](#)
Specifies a keyword to assign to PDF documents that are generated by the dcpdf_MakePDFDoc action.
- [dcpdf_SetProducer](#)
Specifies the Producer property for PDF documents that are generated by the dcpdf_MakePDFDoc action.
- [dcpdf_SetSubject](#)
Specifies the Subject property for PDF documents that are generated by the dcpdf_MakePDFDoc action.
- [dcpdf_SetTitle](#)
Specifies the Title property for PDF documents that are generated by the dcpdf_MakePDFDoc action.
- [dcpdf_UseAltConversionMethod](#)
Causes dcpdf_CreateTiffFromPDF to use an alternate conversion algorithm.

Parent topic: [Global actions](#)

dcpdf_CreateTiffFromPDF

Converts each page of a PDF file into a TIFF file.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_CreateTiffFromPDF ( )
```

Parameters

None.

Returns

False, if the action is not run at the Batch level; if there are no PDF files in the batch; or if an error occurs while TIFF files are being created. Otherwise, True.

Level

Batch only.

Details

This action looks for PDF files in the current batch; creates Image files (.tiff) for each page in the PDF file; and creates one document for the PDF file.

Attention: `dcpdf_MaxSizeToReconvert` can be used to control how this action creates a TIFF file. PDFs are converted to TIFF using a fast algorithm. With some occasional input documents, this may produce images that are large and may not recognize well, causing them to be automatically reconverted to TIFF using a cleaner but slightly slower algorithm. See the help for `dcpdf_MaxSizeToReconvert` for more information.

»

Important: The font mapping file, used by the convert actions to extract images from PDF, uses "Lucida Unicode" as the default font. The Lucida Unicode font is installed by default in the Microsoft Windows operating system, which makes it a reliable default font setting. It is possible that the font `Arial Sans Unicode` might perform better in some instances when converting from PDF to image. You can change the configuration file to use this font, or other fonts, instead of the default font. However, if the specified font is not installed on the machine that is performing the conversion, the text will render as blank on the final image.

To change the font that is used to render a PDF to image, modify the `fontfile.map` file, located in `\Datacap\dcshared\VeryPDF`, as follows:

1. Open `fontfile.map` in any text editor.
2. Change the `defaultwinfont` value from `L_10646.ttf` to `ARIALUNI.TTF`
3. Save the changes and exit.

This example will change the default font from `Lucida Unicode` to `Arial Sans Unicode`.«

»Previous versions of the Convert actions used `Arial Sans Unicode` as the default font. This default was changed in current versions since this font is not installed by default in the Windows operating system. «

Example:

```
dcpdf_CreateTiffFromPDF()
```

When the action in this example encounters a PDF file in the batch, it creates a corresponding Image file (.tiff) and gives the Image file the PDF file's name.

Parent topic: [dcpdf actions](#)

dcpdf_CreateTiffFromPDF_CreateDocs

Converts each page of a PDF file into a TIFF file and creates a runtime hierarchy.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_CreateTiffFromPDF_CreateDocs (strParam)
```

Parameters

The parameter can be True or False.

True: Causes a document hierarchy to be created when TIFF pages are created from a PDF file.

False: A document hierarchy will not be created, making this action operate exactly like dcpdf_CreateTiffFromPDF.

Returns

False, if the action is not run at the Batch level, or if the source PDF file does not have a minimum number of pages (0). Otherwise, True.

Level

Batch Level.

Details

This action converts the pages in a PDF file to TIFFs, like dcpdf_CreateTiffFromPDF. The difference is that if the parameter is True, a document hierarchy will be created for each PDF to group each of the pages from a single PDF into their own document.



Important: The font mapping file, used by the convert actions to extract images from PDF, uses "Lucida Unicode" as the default font. The Lucida Unicode font is installed by default in the Microsoft Windows operating system, which makes it a reliable default font setting. It is possible that the font `Arial Sans Unicode` might perform better in some instances when converting from PDF to image. You can change the configuration file to use this font, or other fonts, instead of the default font. However, if the specified font is not installed on the machine that is performing the conversion, the text will render as blank on the final image.

To change the font that is used to render a PDF to image, modify the `fontfile.map` file, located in `\Datacap\dcshared\VeryPDF`, as follows:

1. Open `fontfile.map` in any text editor.
2. Change the `defaultwinfont` value from `L_10646.ttf` to `ARIALUNI.TTF`
3. Save the changes and exit.

This example will change the default font from `Lucida Unicode` to `Arial Sans Unicode`.«

»Previous versions of the Convert actions used `Arial Sans Unicode` as the default font. This default was changed in current versions since this font is not installed by default in the Windows operating system. «

Example:

```
dcpdf_CreateTiffFromPDF_CreateDocs ("True")
```

In this example, the action uses a primary algorithm to establish a runtime Document for each source document in the PDF file, and assign pages to the runtime Documents according to their placement in the source PDF files.

Parent topic: [dcpdf actions](#)

dcpdf_MakePDFDoc

Creates a PDF document that contains one or more pages of the current document.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_MakePDFDoc(strParam)
```

Parameters

Comma-separated parameters: IncludeFieldText, PageTypes.

The IncludeFieldText parameter can be True or False.

True: stores recognized field data that is associated with the TIFF image of each page in the PDF file to make a searchable PDF. The text data is not visible, but it is searchable.

False: inserts the image of a page into the PDF file, the recognized text is not inserted.

The PageTypes parameter is a list of one or more page types to include in the PDF output. The page types in the list are separated by commas.

If no parameters are provided, IncludeFieldText defaults to False and all document pages are included in the PDF.

If only the IncludeFieldText parameter is provided, all document pages are included in the PDF.

Smart parameters are supported.

Returns

False if the action is not run at the Document level; if there are no pages in the document; or if an error occurs while the PDF file is being created. Otherwise, True. If there are pages in the document but none of them match the specified page type, the action does not create a PDF file and still returns True.

Level

Document Level.

Details

Creates a PDF document that contains one or more pages of a document. The PDF format can contain only the original image or it can contain the image along with the recognized text to make the PDF searchable.

Note: The "dcpdf_MakePDFDoc" action supports only the English language.

Only zoned field text is included if the input parameter is True and is searchable within the PDF. The searchable text might not accurately reflect the text position within the original input image. If you need the entire page

text to be included as searchable text, or more accurate alignment, use one of the recognition actions to create the PDF, such as `RecognizeDocToPDF` from the `OCR_S` library.

To exclude specific page types, set the variable `typesToExclude` to a comma delimited list of page types to exclude from the PDF.

To include specific page types, set the variable `typesToInclude` to a comma delimited list of page types to include in the PDF.

To exclude specific page status, set the variable `statusToExclude` to a comma delimited list of page status to exclude from the PDF.

When more than one filter is specified, the following order of precedence takes place:

- `statusToExclude` overrides `typesToInclude`
- `typesToInclude` overrides `typesToExclude`

If you are calling the action at the Document level, the types and status filters apply to both the documents and their child pages.

If you are calling the action at the Page level, the types and status filters apply to the page only.

These variables must be set before you call the `RecognizeToPDFOCR_A` action.

Example:

```
rrSet ("75", "@D.statusToExclude)
rrSet ("Blank", "@D.typesToExclude)
dcpdf_SetTitle ("MedicalClaims")
dcpdf_SetSubject ("Validated")
dcpdf_SetAuthor ("Steven Moffat")
dcpdf_SetProducer ("Russell Davies")
dcpdf_SetApplication ("MClaims")
dcpdf_MakePDFDoc ("True")
```

This example sets several properties of the PDF file and creates a PDF that contains the recognized text along with all of the page images for the current document. Pages whose type is "Blank" and status is "75" are skipped.

```
dcfpdf_MakePDFDoc ("False,Main_Page")
```

This example does not include the field text and exports only the page types of type `Main_Page`.

```
dcfpdf_MakePDFDoc ("True,Main_Page,Trailing_Page")
```

This example includes the field text and exports only the page types of type `Main_Page` or `Trailing_Page`.

Parent topic: [dcpdf actions](#)

dcpdf_MaxSizeToReconvert

Causes the `dcpdf_CreateTiffFromPDF` action to use a different conversion algorithm if the file that results from the default algorithm exceeds the specified size.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_MaxSizeToReconvert (strParam)
```

Parameters

A Numeric value representing the maximum image size in KB before attempting to convert with an alternate algorithm. If the value is 0, then the alternate algorithm will never be used.

Returns

False if the parameter is not Numeric. Otherwise, True.

Level

Batch or Document level.

Details

Causes `dcpdf_CreateTiffFromPDF` to use a different algorithm based on the configured file size.

The conversion of PDF to TIFF usually works fast and produces clean images. From time to time, the resulting TIFF may become unusually large. A reason for this has been due to the background becoming a dithered light gray background instead of a pure white. This leads to very large image sizes because compression is now less efficient. It also could lead to reduced recognition quality.

To compensate for these rare situations, this action can be used to produce a clean TIFF from a PDF that compresses well. When a TIFF is created from a PDF, the file size is checked to see if it exceeds the value set by this action. If the file size is smaller, then is considered a successful conversion. If the file size is larger, then this alternate method is performed.

This action must be called prior to `dcpdf_CreateTiffFromPDF`. If this action is not called, the default value of 2000KB is used. Calling `dcpdf_SetImageGrayscale(TRUE)` will cause `dcpdf_CreateTiffFromPDF(True)` to always use the alternate algorithm.

Example:

```
dcpdf_SetMaxImageSize("10000")
dcpdf_CreateTiffFromPDF("True")
```

This example will cause the alternate conversion algorithm to be used if the initial conversion produces a TIFF image that is larger than 10000KB.

Parent topic: [dcpdf actions](#)

dcpdf_SetApplication

Specifies the Application ID property for PDF documents that are generated by the `dcpdf_MakePDFDoc` action.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_SetApplication (strParam)
```

Parameters

String value of the Application ID.

Returns

Always True.

Level

Batch or Document level.

Details

This sets the Application ID property of a PDF document generated by a subsequent `dcpdf_MakePDFDoc` action.

If this action is not called, the value will default to "Application".

Example:

```
dcpdf_SetApplication("Invoices")  
dcpdf_SetAuthor("Harriet Jones")  
dcpdf_MakePDFDoc("True")
```

Parent topic: [dcpdf actions](#)

dcpdf_SetAuthor

Specifies the Author property for PDF documents that are generated by the `dcpdf_MakePDFDoc` action.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_SetAuthor (strParam)
```

Parameters

String value of the Author.

If you do not call this action, the value will default to Should be a Task Name.

Returns

Always True.

Level

Batch or Document level.

Details

This action attaches the Author's name (or a related value) to a PDF page or document generated by a subsequent `dcpdf_MakePDFDoc` action.

Example:

```
dcpdf_SetAuthor("Harriet Jones")
dcpdf_SetProducer("Russell Davies")
dcpdf_SetApplication("Invoices")
dcpdf_MakePDFDoc("True")
```

Parent topic: [dcpdf actions](#)

dcpdf_SetImageBitcount

Sets the bit count, bits per pixel, for images in the PDF document that is generated by the `dcpdf_MakePDFDoc` action.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_SetImageBitcount (strParam)
```

Parameters

Numeric value of the BitCount of any image in a PDF document. The action's acceptable values are:

- 1 for Black and White images (a typical value).
- 8 for Grayscale images.
- 24 for Color images.

Returns

False, if the parameter is not Numeric. Otherwise, True.

Level

Batch or Document level.

Details

Sets the BitCount of any image identified in a PDF file by an action such as `CreateTiffFromPDF`. The value of the parameter for the action typically reflects the nature of the images in the PDF file. The default value is 1 (Black and white images).

Attention: The BitCount is the standard number of bits per pixel throughout an image.

Example:

```
dcpdf_SetImageBitCount ("1")                black and white
images.
dcpdf_SetImageBitCount ("8")
```

```
grayscale images.  
dcpdf_SetImageBitCount ("24")
```

```
color images.
```

Parent topic: [dcpdf actions](#)

dcpdf_SetImageCompression

Specifies the compression method to use when you convert a PDF file to TIFF.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_SetImageCompression (strParam)
```

Parameters

A numeric value representing one the following Compressions:

- 1 = NONE (DUMP MODE)
- 2 = CCITTRLE (CCITT modified Huffman RLE)
- 3 = CCITTFAX3 (CCITT Group 3 fax encoding)
- 4 = CCITTFAX4 (CCITT Group 4 fax encoding)
- 5 = COMPRESSION_LZW (Lempel-Ziv and Welch)
- 7 = JPEG (%JPEG DCT compression)
- 32773 = PACKBITS (Macintosh RLE)

Returns

False, if the parameter is not *Numeric*. Otherwise, True.

Level

Batch or Document level.

Details

Sets the compression that will be used when converting a page from a PDF file to a TIFF file. Group 4 Fax is the most common compression used for text recognition. It produces a lossless compressed black and white image. The default compression value is 4 (CCITT Group 4 fax encoding).

Example:

```
dcpdf_SetImageCompression ("7")  
dcpdf_CreateTiffFromPDF ("True")
```

Parent topic: [dcpdf actions](#)

dcpdf_SetImageGrayscale

Specifies how gray areas of a grayscale image are handled when you convert a PDF file to TIFF.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_SetImageGrayscale (strParam)
```

Parameters

The parameter can be True or False.

- True: Gray areas of the image must not be dithered.
- False: Gray areas of the image must be dithered. This option is the default.

Returns

False, if the parameter is invalid. Otherwise, True.

Level

Batch or Document.

Details

Sets a Grayscale flag for images in a PDF file that is generated by an action such as `dcpdf_CreateTiffFromPDF`.

This action controls how gray areas of a grayscale image are handled when converted to black and white. For example, if you have a text document with a gray background, it is recommended to call this action and pass True. This option causes the gray area below a predefined tolerance to be converted to white, producing an image that can be recognized. If you have a grayscale image and this action is set to False, gray areas are dithered to simulate gray in the resulting TIFF image. The default value is False.

If this action is used, it must be called before `dcpdf_CreateTiffFromPDF` or `dcpdf_CreateTiffFromPDF_CreateDocs`.

Example:

```
dcpdf_SetImageGrayscale ("True")  
dcpdf_CreateTiffFromPDF ("True")
```

Parent topic: [dcpdf actions](#)

dcpdf_SetImageQuality

Specifies the image quality to use when you convert a PDF file to TIFF.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_SetImageQuality (strParam)
```

Parameters

Numeric value, between 1 and 100, of the Image Quality standard for images in the PDF file.

Returns

False, if the parameter is not Numeric. Otherwise, True.

Level

Batch or Document level.

Details

This action determines the resulting image quality when creating a TIFF from a PDF file. A higher number will produce a better looking image, but may require more processing time. The default value is 100.

Example:

```
dcpdf_SetImageQuality("75")
dcpdf_CreateTiffFromPDF("True")
```

Parent topic: [dcpdf actions](#)

dcpdf_SetImageResolution

Specifies the output resolution to use when you convert a PDF file to TIFF.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_SetImageResolution (strParam)
```

Parameters

Two comma-separated Numeric values specifying X resolution and Y resolution (in that order). The values are expressed in Dots (pixels) Per Inch (DPI).

Returns

False, if the parameters are not Numeric. Otherwise, True.

Level

Batch or Document level.

Details

This action sets the X and Y resolution for the pages of a PDF that is converted to TIFF. If this action is not called, the default of 200 x 200 will be used. It is strongly recommended that the X and Y resolutions are

always set the same to produce an isotropic image, which will allow for better fingerprinting and recognition.

This action must be called before the `dcpdf_CreateTiffFromPDF` or `dcpdf_CreateTiffFromPDF_CreateDocs` actions.

Example:

```
dcpdf_SetImageResolution("300,300")  
dcpdf_CreateTiffFromPDF()
```

Parent topic: [dcpdf actions](#)

dcpdf_SetKeywords

Specifies a keyword to assign to PDF documents that are generated by the `dcpdf_MakePDFDoc` action.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_SetKeywords (strParam)
```

Parameters

String value of the Keyword.

If you do not call this action, the keyword value is left blank.

Returns

Always True.

Level

Batch or Document level.

Details

This action assigns a single Keyword to a PDF page or document that is generated by a subsequent `dcpdf_MakePDFDoc` action.

Use this action repeatedly within a rule to assign more Keywords.

Important: This action must precede the `dcpdf_MakePDFDoc` action.

Example:

```
dcpdf_SetKeywords("Invoices")  
dcpdf_MakePDFDoc("True")
```

Parent topic: [dcpdf actions](#)

dcpdf_SetProducer

Specifies the Producer property for PDF documents that are generated by the `dcpdf_MakePDFDoc` action.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_SetProducer (strParam)
```

Parameters

String value of the Producer ID.

Returns

Always True.

Level

Batch or Document level.

Details

Sets the producer value of the PDF document. This action must be called prior to calling `dcpdf_MakePDFDoc`. If this action is not called, the value will default to Producer.

Example:

```
dcpdf_SetAuthor("Steven Moffat")
dcpdf_SetProducer("Russell Davies")
dcpdf_SetApplication("Invoices")
dcpdf_MakePDFDoc("True")
```

Parent topic: [dcpdf actions](#)

dcpdf_SetSubject

Specifies the Subject property for PDF documents that are generated by the `dcpdf_MakePDFDoc` action.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_SetSubject (strParam)
```

Parameters

String value of the Subject.

Returns

Always True.

Details

This action sets the Subject property of PDF generated by a subsequent `dcpdf_MakePDFDoc` action.

The Subject is a searchable value for the page or document. If this action is not called, the value of the subject is blank in the resulting document.

Example:

```
dcpdf_SetSubject("HealthClaimDoc")
dcpdf_SetAuthor("Harriet Jones")
dcpdf_MakePDFDoc("True")
```

Parent topic: [dcpdf actions](#)

dcpdf_SetTitle

Specifies the Title property for PDF documents that are generated by the `dcpdf_MakePDFDoc` action.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_SetTitle (strParam)
```

Parameters

String value of the Title.

Returns

Always True.

Level

Batch or Document level.

Details

This action sets the Title property of a PDF document generated by a subsequent `dcpdf_MakePDFDoc` action.

The Title is a searchable value for the page or document. If this action is not called prior to `dcpdf_MakePDFDoc`, the value will default to Untitled.

Example:

```
dcpdf_SetTitle("NewInvoice")
dcpdf_MakePDFDoc("True")
```

Parent topic: [dcpdf actions](#)

dcpdf_UseAltConversionMethod

Causes dcpdf_CreateTiffFromPDF to use an alternate conversion algorithm.

Member of namespace

dcpdf

Syntax

```
bool dcpdf_UseAltConversionMethod ()
```

Parameters

None.

Returns

Always True.

Level

Batch or Document level.

Details

There are two internal algorithms that are used to convert a PDF to a TIFF. This action enables the alternate algorithm. It is recommended that the alternate algorithm is used. Testing has shown that it produces cleaner TIFF images.

This action must be called prior to the dcpdf_CreateTiffFromPDF or dcpdf_CreateTiffFromPDF_CreateDocs actions.

Example:

```
dcpdf_UseAltConversionMethod()  
dcpdf_CreateTiffFromPDF()
```

Parent topic: [dcpdf actions](#)

DocumentAnalytics actions

Use the DocumentAnalytics actions to identify the content type of text blocks and extract data elements from the page layout file.

The DocumentAnalytics actions take advantage of the document's layout and font attributes to use analytics to extract data. These actions depend on a page or document level layout XML file such as tm000001._layout.xml. You use the OCR_A or OCR_S Recognize action to create a layout, then call DocumentAnalytics actions to identify the type of each block (AnalyzeLayout) and to extract data elements from the text.

The layout XML file groups text into blocks as a person would looking at the document. Each block can either have the default type of block or a specific type such as *title* or *table*. There are *Locate* actions available to

navigate the block structure (for example, `GoSiblingBlockNext`). This capability is in contrast to the CCO file produced by other actions that groups text into lines that span the width of the page. The layout XML file also retains font and color attributes, saved in CSS format, for the text. These retained attributes are used for extracting data and reconstructing the document in a new format.

Actions that can produce the layout XML include `OCR_SR.Recognize` and `OCR_A.Recognize`, both of which can process color images and PDF files. To use the *Locate* actions and perform *click 'n' key* during verification, use the `CreateCcoFromLayout` action from the `SharedRecognitionTools` library to create a CCO file for the page after producing the layout XML file.

- [FindLabelValuePairs](#)
Finds label value pairs in the document or page. Use `FindLabelValuePair` to extract a value after calling this action.
- [FindLabelValuePair](#)
Updates the target by finding a value that uses a regular expression to match its label.
- [CopyLabelValuePairs](#)
Copies all found pairs of labels and values to new fields in the DCO object.
- [CreateHTML](#)
Creates an HTML document based on the text and format captured in the layout XML file
- [CopyAllBlocks](#)
Copies all found blocks to DCO fields.
- [AnalyzeLayout](#)
Merges and splits blocks into logical groupings. Sets blocks to more specific types such as Title, Header, or Footer.
- [FindPatterns](#)
Finds patterns in the document or page.
- [ExtractText](#)
Finds entities such as names and addresses in the text by using text analytics. The results are saved and can then be used by subsequent actions, such as `FindExtractedText`.
- [ExtractTextAlchemyLanguage \(deprecated\)](#)
The action calls the AlchemyAPI Combined Call feature and passes in text from a single page at a time. The results are saved as entities in the page's layout XML file. Use the `FindExtractedText` action to populate fields with results of this action.
- [ExtractTextLogEnable](#)
Call this action to enable logging to the specified path and file name
- [ExtractTextNLP](#)
The action calls the Natural Language Processing (NLP) API and passes in text from a single page at a time. The results are saved as entities in the page's layout XML file. Use the `FindExtractedText` action to populate fields with results of this action. To extract from non-English-language text, set the page variable `hr_locale` to the wanted language before calling this action. For example, for the Japanese language, call `rrset("ja","@P.hr_locale")`.
- [FindExtractedText](#)
Finds patterns in the document or page.

Parent topic: [Global actions](#)

Related reference:

[GetSelectedBlockType](#)

[GoSiblingBlockNext](#)

[GoSiblingBlockPrevious](#)

[IsSelectedBlockType](#)

[RegExFind_InBlock](#)

[RegExFindNext_InBlock](#)

[SelectParentBlock](#)

[SelectParentBlockOuterType](#)

[SelectParentBlockType](#)
[UpdateFieldWithBlock](#)
[Recognize](#)
[Recognize](#)

FindLabelValuePairs

Finds label value pairs in the document or page. Use FindLabelValuePair to extract a value after calling this action.

Syntax

```
bool FindLabelValuePairs ()
```

Parameters

None.

Returns

True.

Level

Page level

Details

FindLabelValuePairs analyzes the document or page, finding pairs of labels and their values, which can subsequently be used by other actions to populate fields. The label and values pairs are determined using font attributes and relative positioning of labels to values to identify what is a label and what is a value.

The prepared results can be used by FindLabelValuePair or CopyLabelValuePairs and can be used instead of or in conjunction with locate actions. The results are also saved to the layout XML file, allowing an application developer to easily review the identified pairs and then determine how they should be used within the application.

For example, if the document contains `Total: $100.00`, a label value pair will be created with the label `Total` and the value `$100`. You can then use the FindLabelValuePair action to populate specific fields with found values. You can call the CopyLabelValuePairs action to create a field and value for each match found.

FindLabelValuePairs requires a previously created layout file (for example: `tm000001_layout.xml`) where text is grouped into blocks. See [DocumentAnalytics actions](#) for information on the layout XML file.

Example

```
Recognize()  
  
FindLabelValuePairs()  
  
FindLabelValuePair(Total,@P\Total
```

Parent topic: [DocumentAnalytics actions](#)

FindLabelValuePair

Updates the target by finding a value that uses a regular expression to match its label.

Syntax

```
bool FindLabelValuePair (string Regex , string Target)
```

Parameters

string Regex

Smart parameter for the regular expression to find a matching label. For example, to search for the label `Total`, use `Total`.

string Target

Smart parameter for the target object to save the value to.
If you are using this action on a field, use `@F` to update the current field.

Returns

True if the keyword is found and the following node is a table or if the keyword is found in a table cell.
Otherwise, False.

Level

Page or Field level.

Details

Populates a field using a value that is found by using the FindLabelValuePairs action. The label and values are distinguished by using font attributes and relative positioning of labels to values to identify what is a label and what is a value. FindLabelValuePair is used instead of or in conjunction with *Locate* actions.

FindLabelValuePair requires a previously created layout file (for example: `tm000001_layout.xml`) where text is grouped into blocks. See [DocumentAnalytics actions](#) for information on the layout XML file.

Example

```
Recognize ()  
FindLabelValuePairs ()  
FindLabelValuePair (Total, @P\Total)
```

Parent topic: [DocumentAnalytics actions](#)

CopyLabelValuePairs

Copies all found pairs of labels and values to new fields in the DCO object.

Syntax

```
bool CopyLabelValuePairs ()
```

Parameters

None.

Returns

True.

Level

Page level.

Details

After calling `FindLabelValuePairs`, this action creates fields in the batch hierarchy for each pair that was found. The field ID is the identified label combined with a unique number (for example: `Last Name-1`). The field type is the block type (for example: `block` or `title`). The font and color attributes are saved in CSS format to the style variable. Use `CopyLabelValuePairs` during development to see if the `FindLabelValuePairs` and `FindLabelValuePair` actions can be used to find the fields to be extracted. You can also use this action to create an application where the unwanted fields will be deleted.

`CopyLabelValuePairs` requires a previously created layout file (for example: `tm000001_layout.xml`) where text is grouped into blocks. See [DocumentAnalytics actions](#) for information on the layout XML file.

Example

```
Recognize ()
```

```
FindLabelValuePairs ()
```

```
CopyLabelValuePairs ()
```

Parent topic: [DocumentAnalytics actions](#)

CreateHTML

Creates an HTML document based on the text and format captured in the layout XML file

Syntax

```
bool CreateHTML ()
```

Parameters

None

Returns

True if the HTML document is created. Otherwise, False.

Level

Page level

Details

Creates an HTML document using the text and format in the layout XML file. This is helpful in cases where a well-formatted HTML representation of the document is useful for processing or viewing the document in another product after capture is complete.

The HTML file name is constructed from the ID of the Page (the current object when the action is invoked), followed by the .html file extension. Images and other graphics in the original document are not included in this HTML file.

CreateHTML requires a previously created layout file (for example: tm000001_layout.xml) where text is grouped into blocks. See [DocumentAnalytics actions](#) for information on the layout XML file.

Example

```
Recognize ()
```

```
CreateHTML ()
```

Parent topic: [DocumentAnalytics actions](#)
»

CopyAllBlocks

Copies all found blocks to DCO fields.

Syntax

```
bool CopyAllBlocks ()
```

Parameters

None.

Returns

True.

Level

Page level.

Details

Creates fields in the batch document hierarchy for each block of text. The field ID is the type of block that was found combined with a unique number (for example: `block-1`). The field type is the block type (for example: `block` or `title`). The font and color attributes are saved in CSS format to the style variable. The text contents of the block will be the field's text value.

Entities found using the [FindExtractedText](#), [FindLabelValuePairs](#) and [ExtractTextAlchemyLanguage \(deprecated\)](#) actions are copied as new fields. AlchemyAPI entities fields are saved as field variables unless the field name is text.

CopyAllBlocks requires a previously created layout file (for example: tm000001_layout.xml) where text is grouped into blocks. See [DocumentAnalytics actions](#) for information on the layout XML file.

Example

```
Recognize()
```

```
CopyAllBlocks()
```

Parent topic: [DocumentAnalytics actions](#)
«

AnalyzeLayout

Merges and splits blocks into logical groupings. Sets blocks to more specific types such as Title, Header, or Footer.

Syntax

```
bool AnalyzeLayout ()
```

Parameters

None.

Returns

True.

Level

Page level

Details

AnalyzeLayout improves the grouping of text into blocks and changes the type of block to be more specific (for example: Changes the type from block to title). This action can be helpful for extracting data with FindLabelValuePairs where the value to be extracted spans multiple lines.

AnalyzeLayout requires a previously created layout file (for example: tm000001_layout.xml) where text is grouped into blocks. See [DocumentAnalytics actions](#) for information on the layout XML file.

Table 1. Block types vs XML node

Block type	Node in the layout XML file
Block	Block
Header	Header
Footer	Footer
Title	Title
Heading1	H1
Heading2	H2

Block type	Node in the layout XML file
Heading3	H3
Picture	Picture
Barcode	Barcode
Space	S
Tab	Tab
Table	Table
Row	Row
Cell	Cell
Paragraph	Para
Line	L
Sentence	Sent
Word	W
Character	C

After you identify the blocks, you can use the following *Locate* actions to navigate and select the blocks:

- GetSelectedBlockType
- GoSiblingBlockNext
- GoSiblingBlockPrevious
- IsSelectedBlockType
- RegExFind_InBlock
- RegExFindNext_InBlock
- SelectParentBlock
- SelectParentBlockOuterType
- SelectParentBlockType
- UpdateFieldWithBlock

Example

```
Recognize ()
```

```
AnalyzeLayout ()
```

Parent topic: [DocumentAnalytics actions](#)

FindPatterns

Finds patterns in the document or page.

Restriction: This action does not support regular expressions containing a line break nor tab expressions.

Syntax

```
bool FindPatterns (string patternsFilePath)
```

Parameters

patternsFilePath
Path to XML file containing patterns to match.

Returns

True.

Level

Page level

Details

Analyzes all blocks of text to determine if addresses, dates, or custom expressions are present. Regular expressions are stored in an XML file as a list of patterns with the properties listed at the end of this topic. Each pattern must have a unique `id` attribute. The pattern `type` attribute is the DCO field to populate. The pattern `value` is the regular expression.

FindPatterns requires a previously created layout file (for example: tm000001_layout.xml) where text is grouped into blocks. See [DocumentAnalytics actions](#) for information on the layout XML file.

Example

```
Recognize()  
  
FindPatterns("@APPVAR(values/gen/patternsPath)")
```

Format of a patter in the pattern XML file

```
<Pattern  
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcacb148_un  
iqueString" type="documentHierarchyFieldType" enabled="true">  
  regularExpression  
</Pattern>
```

Example of a pattern XML file

```
<Patterns>  
<Pattern  
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcacb148_ad  
dressPattern1" type="us_address" enabled="true">  
  (\d{1,5}.\{1,16} (Alley|Avenue| (Ave\.\?) | (Bvd\.\?) |Blvd|Boulevard|Circle| (Cir\.\?)  
  |Street| (St\.\?) | ([P]\.\?\s*?[O]\.\?\s*?Box) |Drive| (Dr\.\?) | (Cres\.\?) |Crescent|Court|  
  (Ct\.\?)  
  |Way| (Tr\.\?) |Terrace|Trail| (Rd\.\?) |Road|Lane|Highway| (Hwy\.\?) | (Apt\.\?) |  
  (Pl\.\?) |Place) .*?  
  (?:  
  (A[KLRZ] |C[AOT] |D[CE] |FL|GA|HI|I[ADLN] |K[SY] |LA|M[ADEINOST] |N[CDEHJMVY] |O[HKR] |P[AR]  
  |RI|S[CD] |T[NX] |UT|V[AIT] |W[AIVY] ) ) |  
  (Alabama|Alaska|Arizona|Arkansas|California|Colorado  
  
  |Connecticut|Delaware|Florida|Georgia|Hawaii|Idaho|Illinois|Indiana|Iowa|Kansas|Kent  
  ucky  
  
  |Louisiana|Maine|Maryland|Massachusetts|Michigan|Minnesota|Mississippi|Missouri|Mont  
  ana  
  |Nebraska|Nevada|New\s? (Hampshire|Jersey|Mexico|York) |North\s?  
  (Carolina|Dakota) |Ohio|Oklahoma
```

```

|Oregon|Pennsylvania|Rhode\s?Island|South\s?
(Carolina|Dakota)|Tennessee|Texas|Utah|Vermont
|Virginia|Washington|West\s?Virginia|Wisconsin|Wyoming)|
(ALABAMA|ALASKA|ARIZONA|ARKANSAS

|CALIFORNIA|COLORADO|CONNECTICUT|DELAWARE|FLORIDA|GEORGIA|HAWAII|IDAHO|ILLINOIS|INDI
ANA|IOWA

|KANSAS|KENTUCKY|LOUISIANA|MAINE|MARYLAND|MASSACHUSETTS|MICHIGAN|MINNESOTA|MISSISSIP
PI|MISSOURI
|MONTANA|NEBRASKA|NEVADA|NEW\s?(HAMPSHIRE|JERSEY|MEXICO|YORK)|NORTH\s?
(CAROLINA|DAKOTA)|OHIO
|OKLAHOMA|OREGON|PENNSYLVANIA|RHODE\s?ISLAND|SOUTH\s?(CAROLINA|
DAKOTA)|TENNESSEE|TEXAS|UTAH
|VERMONT|VIRGINIA|WASHINGTON|WEST\s?VIRGINIA|WISCONSIN|WYOMING))\s*\d{5}((\s*
|\s*)\d{4})?)
</Pattern>
<Pattern
id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcacb148_da
tePattern1" type="date" enabled="true">
((?:J(anuary|u(ne|ly))|February|Ma(rch|y)|A(pril|ugust)|
((Sept|Nov|Dec)em)|October)
| (Jan|Feb|Mar|Apr|May|Aug|Sep|Sept|Oct|Nov|Dec)) (\s*|\s-)\d{1,2}\,?\s*|\s-)\d{4})
|(\d{2}\/\d{2}\/\d{4})|(\d{2}th\s*
(?:J(anuary|u(ne|ly))|February|Ma(rch|y)|A(pril|ugust)
|((Sept|Nov|Dec)em)|October)| (Jan|Feb|Mar|Apr|May|Aug|Sep|Sept|Oct|Nov|Dec))
[\s*\,]\d{4})
</Pattern>
</Patterns>

```

Parent topic: [DocumentAnalytics actions](#)

ExtractText

Finds entities such as names and addresses in the text by using text analytics. The results are saved and can then be used by subsequent actions, such as FindExtractedText.

Restriction: This action does not support regular expressions that contain a line break nor tab expressions.

Syntax

```
bool ExtractText (string extractors)
```

Parameters

extractors
Smart parameter for a comma-separated list of extractors to process.

Returns

True.

Level

Document or Page level

Details

Finds entities such as names and addresses in the text by using text analytics. The results are saved and can then be used by subsequent actions, such as FindExtractedText. To extract from non-English text, set the page variable `hr_locale` to the desired language before calling this action. For example, for Japanese call `rrset("ja", "@P.hr_locale")`.

The entities that are found are determined by AQL extractors. An initial set of pre-built extractors are provided and while they work in many instances they may not work in every case. See the IBM BigInsights documentation for pre-built extractors. Additional extractors may be created using the IBM BigInsights tools for creating AQL extractors.

Extractors are saved in compiled files with the extension `tam`. All `tam`, dictionary, and table files present in the `\rrs\aqf` folder will be loaded. The extractors provided by Datacap are exposed in `DatacapPreBuilt_BasicFeatures.tam`. You can add or remove `tam` files to the `\rrs\aqf` folder to control if they are executed or not.

All of the loaded extractors are run on the document or page.

Important: The results from ExtractText are saved to the layout XML file, for example `tm000001_layout.xml`, which can be opened in a text editor to see the available entities and the entity fields that can be copied into document hierarchy fields. You can search for an entity in the layout XML using its name such as `Address.Address`.

Important: This action requires a 32-bit Java runtime environment. The default location is `\Datacap\dcshared\jre` or else the path specified in the `JAVA_HOME` system variable is used.

ExtractText requires a previously created layout file (for example: `tm000001_layout.xml`) where text is grouped into blocks. See [DocumentAnalytics actions](#) for information on the layout XML file.

Example

The following example populates the city field with the first instance of an address where the state is California.

```
ExtractText (DateTime.DateTime, Address.Address)
FindExtractedText (@P\City, First, Address.Address, city, stateorprovince, (California) |
(CA))
```

Support for external dictionaries

The ExtractText action supports AQL external dictionary. Using this feature, you can write annotators that do not need to be recompiled when a change is needed.

You can export from the IBM® InfoSphere® BigInsights web tools and place the exported folders into `\rrs\aqf\src` location. The AQL is compiled at the run time.

Note: You need to manually copy any external dictionaries and tables to the `\rrs\aqf` location.

It is recommended that you keep a back up of RRS folder, in case a file or folder gets corrupted at the time of copying or due to misconfiguration.

Detailed steps to configure Custom annotators in Datacap:

Complete the following steps to configure Custom annotators in Datacap.

1. Once you create Custom Extractor using BigInsights Web tool, export the extractor as "Executables" with an option of including "Source files" . Export in a zip format.
2. Copy TAMs file from export to the `\rrs\aqf` folder. Do NOT copy the `InputDocumentProcessor.TAM` file. Leave the original in `\rrs\aqf` folder itself.
3. Copy the SRC folder from Export (one from exported zip) folder to `\rrs\aqf`.

4. Make sure to copy all the supported *.DICT files provided by BigInsights in rrs\aqf folder.

Custom extractors must be called with ExtractText action in the format Module.Viewname.

Verify the modulename from corresponding aql file.

For e.g. ZIPCODE_BasicFeatures.ZCView

After the compilation process completes, the compiled TAM files are saved in \rrs\aqf location. Ensure that you remove the rrs\aqf\src folders after the compilation process.

List of pre-built extractors

The following Datacap extractor names consist of the following two parts, separated by a period: the InfoSphere BigInsights extractor name followed by the InfoSphere BigInsights attribute name.

See the IBM InfoSphere BigInsights documentation for more information about pre-built extractors.

http://www.ibm.com/support/knowledgecenter/SSPT3X_3.0.0/com.ibm.swg.im.infosphere.biginsights.text.doc/doc/ana_txtan_extractor-libraries.html

Address.Address
City.City
Continent.Continent
Country.Country
Date.Dates
DateTime.DateTime
EmailAddress.EmailAddress
Facility.Facility
FinancialAnnouncements.CompanyEarningsAnnouncement
FinancialAnnouncements.AnalystEarningsEstimate
FinancialAnnouncements.CompanyEarningsGuidance
FinancialEvents.Alliance
FinancialEvents.Acquisition
FinancialEvents.JointVenture
FinancialEvents.Merger
Location.Location
NotesEmailAddress.NotesEmailAddress
Organization.Organization
Person.Person
PhoneNumber.PhoneNumber
StateOrProvince.StateOrProvince
URL.URL WaterBody.WaterBody
ZipCode.ZipCode

BigInsightsChineseNER.PersonChinese;
BigInsightsChineseNER.LocationChinese;
BigInsightsChineseNER.OrganizationChinese;

BigInsightsJapaneseNER.PersonJapanese;
BigInsightsJapaneseNER.LocationJapanese;
BigInsightsJapaneseNER.OrganizationJapanese;

Parent topic: [DocumentAnalytics actions](#)



ExtractTextAlchemyLanguage (deprecated)

The action calls the AlchemyAPI Combined Call feature and passes in text from a single page at a time. The results are saved as entities in the page's layout XML file. Use the FindExtractedText action to populate fields

with results of this action.

Important: This action is deprecated, and is scheduled to be removed in the future release. To read the announcement from IBM about retirement of *AlchemyAPI*, see <https://www.ibm.com/blogs/bluemix/2017/03/bye-bye-alchemyapi/>. According to the announcement, you no longer be able to provision new *AlchemyAPI* instances after April 7, 2017. However, all existing instances continue to be supported by *AlchemyAPI* until March 7, 2018.

Syntax

```
bool ExtractTextAlchemyLanguage (string options)
```

Parameters

- entity
- keyword
- taxonomy
- concept
- relation
- doc-sentiment
- doc-emotion
- dates

Variables

Timeout

The default timeout is 100 seconds. You can specify the timeout by saving a page variable that is called *alchTimeout* and specifying the time in milliseconds.

Save Response

The response from AlchemyAPI can be saved by setting the variable *alchSaveResponse* to 1.

Returns

True if the layout is loaded and there are no errors. Otherwise, False.

Level

Document or page.

Details

Important: You must have an AlchemyAPI license to use this action. This license is obtained by creating an account on bluemix.net and adding the AlchemyAPI service to a workspace. The URL and key are shown on the Service Credentials page. Enter the credentials in Datacap Application Manager:

- Save the URL in your application's general section in a variable called *AlchemyApiUrl*.
Note: The URL must end in `/calls/`. If the *AlchemyApiUrl* is not updated, the default URL is `https://gateway-a.watsonplatform.net/calls/`
- Save the license key in your application's advanced section in a variable called *AlchemyApiKey*.

Entities are named using the format *AlchemyFeature.ResponseNode*. For example, an entity result for a person will have the name *Entity.Person*. Most entity fields will be named using the node returned by AlchemyAPI.

Nested responses are flattened to one level. For example, the disambiguated sub type will be named *subTypes* and the results will be saved as a comma-separated list.

When called on the document or page, the text of the layout XML file specified in the calling object's layout variable will be sent to AlchemyAPI.

The results will be saved to the entities list of the page in the layout XML file.

Example

```
Recognize ()  
AnalyzeLayout ()  
ExtractTextAlchemyLanguage ()  
FindExtractedText ()
```

Parent topic: [DocumentAnalytics actions](#)
«

ExtractTextLogEnable

Call this action to enable logging to the specified path and file name

Syntax

```
bool ExtractTextLogEnable (string Target)
```

Parameters

Target
Smart parameter of the path and file name to log to.

Returns

True.

Level

Any

Details

Enables additional logging during text extraction. This is used for debugging during system configuration. The file will grow until it is deleted.

Example

```
ExtractTextLogEnable (c:\datacap\extract.log)  
ExtractText (DateTime.DateTime, Address.Address)
```

Parent topic: [DocumentAnalytics actions](#)
»

ExtractTextNLP

The action calls the Natural Language Processing (NLP) API and passes in text from a single page at a time. The results are saved as entities in the page's layout XML file. Use the FindExtractedText action to populate fields with results of this action. To extract from non-English-language text, set the page variable `hr_locale` to the

wanted language before calling this action. For example, for the Japanese language, call `rrset("ja","@P.hr_locale")`.

Syntax

```
bool ExtractTextNLP (string options, string model)
```

Parameter text features

No parameter - returns all results

- entities
- keywords
- categories
- concepts
- semantic_roles
- relations

Variables

Timeout

The default timeout is 150 seconds. You can specify the timeout by saving a page variable called `nlpTimeout` and specifying the time in milliseconds.

Save Response

The response from NLPAPI can be saved by setting the variable `nlpSaveResponse` to 1.

Returns

True if the layout is loaded and there are no errors. Otherwise, False.

Level

Document or page.

Details

Important: You must have an enterprise NLP license to use this action. This license is obtained by creating an account on bluemix.net and adding the NLP service to a workspace. The URL and key are shown on the Service Credentials page. Enter the credentials in Datacap Application Manager:

- Save the URL in your application's general section in a variable called `NLPApiUrl`.
Note: If the `NLPApiUrl` is not updated, the default URL is <https://gateway.watsonplatform.net/natural-language-understanding/api/v1/analyze?version=2017-02-27>
- Save the username in your application's gen section in a variable called `NLPApiUsername`.
- Save the password in your application's adv section in a variable called `NLPApiPassword`.

You can deploy Watson Knowledge Studio custom model to override default model.

Entities are named by using the format `NLPFeature.ResponseNode`. For example, an entity result for a person has the name `Entity.Person`. Most entity fields are named by using the node returned by NLPAPI. Nested responses are flattened to one level. For example, the disambiguated sub type is named `subTypes` and the results are saved as a comma-separated list.

When called on the document or page, the text of the layout XML file specified in the calling object's layout variable will be sent to NLPAPI.

The results will be saved to the entities list of the page in the layout XML file.

To populate a field with a result of this action, use the FindExtractedText action.

Example

```
Recognize ()
AnalyzeLayout ()
ExtractTextNLP ()
FindExtractedText ()
```

Parent topic: [DocumentAnalytics actions](#)



FindExtractedText

Finds patterns in the document or page.

Restriction: This action does not support regular expressions containing a line break nor tab expressions.

Syntax

```
bool FindExtractedText (string Target, string Scope, string Entity, string Source,
string SearchField, string SearchRegex)
```

Parameters

Target

Smart parameter for the target object to save the value to. If used on a field use @F to update the current field.

Scope

Smart parameter first|Last|All

Populates the target using the match or matches by using the specified entity type, search field, and search expression. If First is specified, the first match is used. If Last is specified the last match is used. If All is specified a sub field will be created for all matches. All is supported only when the target is a DCO object and is not supported for variables.

Entity

Smart parameter name of the entity to match. For example `Address.Address` for a result from the ExtractText action, or `Person` for a result from the ExtractTextAlchemyLanguage action. The format of the entity name is `Module.OutputView`.

SearchField

Optional. Smart parameter name of the entity field to match. For example `city`. If specified, matching will be limited to the specified field. If the Entity parameter is set to Field, this field's value will be used to update the target.

SearchRegex

Optional. Smart parameter regular expression to search for. To search for a word use `(SearchWord)`.

Returns

True when a match is found. Otherwise, False.

Level

Any

Details

Populates a field or variable with a value found using the `ExtractText` and `ExtractTextAlchemyLanguage` actions.

Important: The results from the `ExtractText` and `ExtractTextAlchemyLanguage` actions will be saved to the layout XML file (for example: `tm000001_layout.xml`), which can be opened in a text editor to see the available entities and the entity fields that can be copied into document hierarchy fields. You can search for an entity in the layout XML using its name such as `Address.Address`, or `Person`.

If the source is Entity: This can be used to create the line item detail structure in the run time hierarchy for capturing a list of items to be exported.

- If the Target has sub fields defined and the sub field types match the entity field names they will be populated.
- If the Target has sub fields and the sub fields have a variable named `entityMap` with a value that is the same as the entity field name in the layout.xml file the sub field of the target will be populated.
- If the Target does not have a sub field mapped to an entity field name a variable will be created using the entity field name.

`FindExtractedText` requires a previously created layout file (for example: `tm000001_layout.xml`) where text is grouped into blocks. See [DocumentAnalytics actions](#) for information on the layout XML file.

ExtractText example 1

This example will populate a page field called `City` with the first address in California.

```
Recognize()
ExtractText (Address.Address)
FindExtractedText (@P\City,First,Address.Address,city,stateorprovince,(California)|
(CA))
```

ExtractText example 2

This example will create a line item for each address found in the document.

The following document hierarchy is expected to be defined:

```
Field AddressDetail
  Field LineItem
    Field City
      Variable entityMap=city
    Field State
      Variable entityMap=stateorprovince

Recognize()
ExtractText (Address.Address)
FindExtractedText (@P\AddressDetail,All,Address.Address,, ,)
```

ExtractText example 3

This example will populate variables `city`, `stateorprovince`, `zip` and `address` on the target field `AddressDetail`.

```
Recognize()
ExtractText (Address.Address)
FindExtractedText (@P\AddressDetail,First,Address.Address,, ,)
```

ExtractTextAlchemyLanguage example 1

This example will populate a page field called Person with the first Person returned by AlchemyAPI.

```
Recognize ()
ExtractTextAlchemyLanguage (entity)
FindExtractedText (@P\Person, First, entity, text, type, Person)
```

Parent topic: [DocumentAnalytics actions](#)

DocumentAnalytics.VisualRecognitionClassifier actions

Visual Recognition Classifier is an IBM Watson service, which can be trained to classify image-based documents. The default instance name: DocumentAnalytics.Undoable-in-transactional.

You can use the default classifier to do the classification, or you can first create and train a custom classifier by using some representative set of training data. The uploaded training data must contain at least two compressed (.zip) files, each containing sample images of a particular classes.

Once the classifier has been trained, you can give it other similar documents and classifier attempts to classify them according to its training. The Classifier returns a confidence score that is associated with the classification.

Properties and methods:

- [VisualRecogClassify](#)
Classifies image by using IBM Watson Visual Recognition API.
- [VisualRecogSetCredentials](#)
Sets the credentials to be used to do the classification.
- [VisualRecogSetMinConfidence](#)
Sets the minimum confidence score for classification matching.
- [VisualRecogSetURL](#)
Updates the new URL to be used to do the classification.
- [VisualRecogTrain](#)
Creates or replaces a Visual Recognition Classifier.

Parent topic: [Global actions](#)

VisualRecogClassify

Classifies image by using IBM Watson Visual Recognition API.

Syntax

```
bool VisualRecogClassify (string ClassifierName)
```

Parameters

string ClassifierName - Name of the Classifier to be used

Returns

True, if the action succeeds. Otherwise, False

Level

Page level.

Details

This action identifies a page by using the Watson™ Recognition technology. This technology analyzes the image-based full page and attempts to find match within the classes that have been defined for the selected classifier. If a match is found, the Page type is populated with the ID of the category that was matched.

If a match is not found, the page type is set to "Other".

When classification is complete, a list of matches and their confidence values are stored in the "MatchingCategoryX" and "MatchingCategoryConfX" variables. The number of matches is stored in the variable "MatchingCategoriesCount".

To run classification without updating the page type, set the variable "UpdateDCOType" to "0" before calling this action. In this case, classification does not update the page type, but the variables that are mentioned above is populated.

Because this action supports image files (.jpg, .jpeg or .png) only. ConvertToJPEG must be called before using the Classify action for other types of files.

The ClassifierName can be a smart parameter.

Example

```
VisualRecogSetURL("@APPVAR(values/gen/url) ")
VisualRecogSetCredentials("@APPVAR(values/adv/VRAPIKey) ")
VisualRecogClassify("@APPVAR(values/gen/VRClassifierName) ")
```

Note: VisualRecogSetURL action is not included if your IBM Visual Recognition token is created before May 2018.

Parent topic: [DocumentAnalytics.VisualRecognitionClassifier actions](#)

VisualRecogSetCredentials

Sets the credentials to be used to do the classification.

Syntax

```
bool VisualRecogSetCredentials (string APIKey)
```

Parameters

string APIKey can be Smart Parameters.

Returns

True, if the action succeeds. Otherwise, False.

Level

Batch level.

Details

This action sets the credentials to be used to do the classification. The APIKey can be Smart Parameters.

Example

```
VisualRecogSetURL("@APPVAR(values/gen/url) ")
VisualRecogSetCredentials("@APPVAR(values/adv/VRAPIKey) ")
VisualRecogClassify("@APPVAR(values/gen/VRClassifierName) ")
```

Note: VisualRecogSetURL action is not included if your IBM Visual Recognition token is created before May 2018.

Parent topic: [DocumentAnalytics.VisualRecognitionClassifier actions](#)

VisualRecogSetMinConfidence

Sets the minimum confidence score for classification matching.

Syntax

```
bool VisualRecogSetMinConfidence (string MinScore)
```

Parameters

string MinScore - Minimum score for classification matching. Valid values are fractional values between zero and one (for example: 0.0 and 1.0).

Returns

True, if the parameter value is between the valid range of zero to one (0.0 and 1.0) Otherwise, False.

Level

All level.

Details

When Classify searches for a classification match, a score between zero (no match) and one (a positive match) is calculated. This action sets the minimum score that a match must be considered a match. Any matches with a score less than the value specified is rejected. With this action, you can control the tolerance for documents matching an existing example.

When setting up the parameter in your application, use the decimal character from the system locale that is defined for the application in the Taskmaster Application Manager. For example, when the decimal character is a period, use a value from 0.0 to 1.0. When the decimal character is a comma, use a value in the range 0,0 - 1,0.

The MinScore can be a Smart Parameter. This action must be called before the Classify action.

Example

```
VisualRecogSetCredentials("@APPVAR(values/adv/VRAPIKey) ")
VisualRecogSetMinConfidence(0.9)
VisualRecogClassify("@APPVAR(values/gen/VRClassifierName) ")
```

Parent topic: [DocumentAnalytics.VisualRecognitionClassifier actions](#)

VisualRecogSetURL

Updates the new URL to be used to do the classification.

Syntax

```
VisualRecogSetURL (string url)
```

Parameters

The URL can be Smart Parameters

Returns

True, if the action succeeds. Otherwise, False

Level

Batch level.

Details

This action updates the new URL to be used to do the classification. IBM Visual Recognition token created after May 2018 must apply this action before `VisualRecogSetCredentials()`.

Example

```
VisualRecogSetURL("@APPVAR(values/gen/url) ")
VisualRecogSetCredentials("@APPVAR(values/adv/VRAPIKey) ")
VisualRecogClassify("@APPVAR(values/gen/VRClassifierName) ")
```

Parent topic: [DocumentAnalytics.VisualRecognitionClassifier actions](#)

VisualRecogTrain

Creates or replaces a Visual Recognition Classifier.

Syntax

```
bool VisualRecogTrain (string ZIPDirectory, string Name, string deleteExisting)
```

Parameters

string ZIPDirectory - Directory where this action stores the zip file with the training data. the zip file has the name of the classifier. This parameter supports SmartParameters.

string Name - Name to give the classifier. This name is used when you later try to classify a page. This parameter supports SmartParameters.

string deleteExisting - Delete the classifier of the same name if it exists. If this flag is set to true, it deletes the classifier of the same name, if it exists. If this flag is set to false and a classifier with the same name exists, this

action adds the training data to the existing classifier. This parameter supports SmartParameters. The type of this parameter is string only to support SmartParameters. Internally, it is treated as a boolean. Specify '1' for true and '0' for false. Any other value is ignored and the action is default to false.

Returns

True, action is successful. Otherwise, False.

Level

Batch level.

Details

This action creates or replaces a Visual Recognition Classifier. Pages of the batch are used as the training data. It is expected that the batch is in a certain format. In particular, the batch needs to be divided into set documents, where the 'Type' associated with the document is class that is created in the Visual Recognition Service. All of pages in that document are treated as training data for that class. So, for example, if you wanted to train the classifier to recognize the main page and the trailing pages of a mortgage application, you would create a batch with two documents: Main_Page and Trailing_Page. Put all of the training page for the main page under the Main_Page document and the trailing pages under Trailing_Page document.

Example

```
VisualRecogSetURL("@APPVAR(values/gen/url) ")
VisualRecogSetCredentials("@APPVAR(values/adv/VRAPIKey) ")
VisualRecogTrain("@APPPATH(runtime)+\+..\+ZIP", "@APPVAR(values/gen/VRClassifierName)", 1)
```

Parent topic: [DocumentAnalytics.VisualRecognitionClassifier actions](#)

DocumentAnalytics.NaturalLanguageClassifier actions

Natural Language Classifier is an IBM Watson service, which can be trained to classify documents or sections of documents. That based on the text that is contained in the document or the section of a document. The default instance name: DocumentAnalytics.NaturalLanguageClassifierActions.

To do this, you must first create and train a classifier using some representative set of training data containing text from a set of sample documents. Once the classifier has been trained, you can give it other similar documents and classifier attempts to classify them according to its training. The Classifier returns a confidence score that is associated with the classification.

Because Natural Language Classifier works on the text of the document, you must first perform OCR on the document before calling the Classify action to do the classification.

Properties and methods:

- [NLCClassify](#)
Identifies a page by using the IBM Natural Language technology.
- [NLCClassifyText](#)
Classifies the specified text by using the IBM Natural Language technology.
- [NLCSetsCredentials](#)
Sets the credentials to be used to do the classification.
- [NLCSetsLanguage](#)
Sets the language of the page to be classified.

- [NLCSetsMinConfidence](#)
Sets the minimum confidence score for classification matching.
- [NLCTrain](#)
Creates or replaces an NLC Classifier.

Parent topic: [Global actions](#)

NLCClassify

Identifies a page by using the IBM Natural Language technology.

Syntax

```
bool NLCClassify (string ClassifierName)
```

Parameters

string ClassifierName - Name of the Classifier to be used.

Returns

True, action is successful. Otherwise, False..

Level

Page level.

Details

This action identifies a page by using the IBM Natural Language technology. This technology analyzes the full text of pages and attempts to find match within the classes that have been defined for the selected classifier. If a match is found, the Page type is populated with the ID of the category that was matched.

If a match is not found, the page type is set to "Other".

When classification is complete, a list of matches and their confidence values are stored in the "MatchingCategoryX" and "MatchingCategoryConfX" variables. The number of matches is stored in the variable "MatchingCategoriesCount" .

To run classification without updating the page type, set the variable "UpdateDCOType" to "0" before calling this action. In this case classification will not update the page type, but the variables that are mentioned above will still be populated.

Because the matching relies on a page's full text, a full page recognition action must be called before using the Classify action.

This action gets the text for the page in the following order:

- from the layout.xml that is file generated by the Recognize action
- from the .txt file that is generated by the RecognizeToFile action
- from the .cco file that is generated by the RecognizePage action

The ClassifierName can be a smart parameter.

Example

```
NLCSetLanguage("en")
NLCSetCredentials("@APPVAR(values/gen/NLCUserName)", "@APPVAR(values/adv/NLCPassword)")
NLCSetMinConfidence(0.9)
Recognize()
NLCClassify("@APPVAR(values/gen/NLCClassifierName)")
```

Parent topic: [DocumentAnalytics.NaturalLanguageClassifier actions](#)

NLCClassifyText

Classifies the specified text by using the IBM Natural Language technology.

Syntax

```
bool NLCClassifyText (string ClassifierName, string TextToClassify)
```

Parameters

string ClassifierName - Name of the Classifier to be used

string TextToClassify - Text to be classified by using the Natural Language Classifier. This parameter supports SmartParameters.

Returns

True, action is successful. Otherwise, False.

Level

Page level.

Details

This action classifies the specified text by using the IBM Natural Language technology. This technology analyzes the text that is specified and attempts to find match within the classes that have been defined for the selected classifier.

When classification is complete, a list of matches and their confidence values are stored in the "MatchingCategoryX" and "MatchingCategoryConfX" variables. The number of matches is stored in the variable "MatchingCategoriesCount".

Both ClassifierName and TextToClassify can be smart parameters.

Example

```
NLCSetCredentials("@APPVAR(values/gen/NLCUserName)", "@APPVAR(values/adv/nlcpassword)")
NLCSetMinConfidence(0.9)
NLCClassifyText("@APPVAR(values/gen/NLCClassifierName)", "Sample Text")
```

Parent topic: [DocumentAnalytics.NaturalLanguageClassifier actions](#)

NLCSetCredentials

Sets the credentials to be used to do the classification.

Syntax

```
bool NLCSetCredentials (string UserName , string Password)
```

Parameters

string UserName - UserName

string Password - Password

Returns

True, if the action succeeds. Otherwise, False.

Level

All level.

Details

This action sets the credentials to be used to do the classification. The UserName and password can be Smart Parameters. In order to protect the secrecy of the Password, it is recommended that you store it as an Advanced custom value in the Datacap Application Manager.

Example

```
NLCSetLanguage ("en")
NLCSetCredentials ("@APPVAR(values/gen/NLCUserName) ", "@APPVAR(values/adv/NLCPassword) ")
NLCSetMinConfidence (0.9)
Recognize ()
NLCClassify ("@APPVAR(values/gen/NLCClassifierName) ")
```

Parent topic: [DocumentAnalytics.NaturalLanguageClassifier actions](#)

NLCSetLanguage

Sets the language of the page to be classified.

Syntax

```
bool NLCSetLanguage (string LanguageCode)
```

Parameters

string LanguageCode - The two-letter code associate with the language of the page to be classified

Returns

True, if the action succeeds. Otherwise, False.

Level

All level.

Details

This action sets the language of the page to be classified. The LanguageCode can be a smart parameter. The possible values for the Language code are:

- 'en' : English
- 'ar' : Arabic
- 'fr' : French
- 'de' : German
- 'it' : Italian
- 'ja' : Japanese
- 'ko' : Korean
- 'pt' : Portuguese (Brazilian)
- 'es' : Spanish

This action must be called before the Classify action.

Example

```
NLCSetLanguage("en")
NLCSetCredentials("@APPVAR(values/gen/NLCUserName)", "@APPVAR(values/adv/NLCPassword)")
NLCSetMinConfidence(0.9)
Recognize()
NLCClassify("@APPVAR(values/gen/NLCClassifierName)")
```

Parent topic: [DocumentAnalytics.NaturalLanguageClassifier actions](#)

NLCSetMinConfidence

Sets the minimum confidence score for classification matching.

Syntax

```
bool NLCSetMinConfidence (string MinScore)
```

Parameters

string MinScore - Minimum score for classification matching. Valid values are fractional values between zero and one (for example: 0.0 and 1.0)

Returns

True, if the parameter value is between the valid range of zero to one (0.0 and 1.0) Otherwise, False.

Level

All level.

Details

When Classify searches for a classification match, a score between zero (no match) and one (a positive match) is calculated. This action sets the minimum score that a match must be considered a match. Any matches with a score less than the value specified is rejected. With this action, you can control the tolerance for documents matching an existing example.

When setting up the parameter in your application, use the decimal character from the system locale that is defined for the application in the Taskmaster Application Manager. For example, when the decimal character is a period, use a value from 0.0 to 1.0. When the decimal character is a comma, use a value in the range 0,0 - 1,0.

The MinScore can be a Smart Parameter. This action must be called before the Classify action.

Example

```
NLCSetLanguage("en")
NLCSetCredentials("@APPVAR(values/gen/NLCUserName)", "@APPVAR(values/adv/NLCPassword)")
NLCSetMinConfidence(0.9)
Recognize()
NLCClassify("@APPVAR(values/gen/NLCClassifierName)")
```

Parent topic: [DocumentAnalytics.NaturalLanguageClassifier actions](#)

NLCTrain

Creates or replaces an NLC Classifier.

Syntax

```
bool NLCTrain (string CSVDirectory, string Name, string deleteExisting, string
appendToCSV)
```

Parameters

string CSVDirectory - Directory where this action stores the csv file with the training data. CSV file has the name of the classifier. This parameter supports SmartParameters.

string Name - Name to give the classifier. This name is used when you later try to classify a page. This parameter supports SmartParameters.

string deleteExisting - Delete the classifier of the same name if it exists. If this flag is set to true, it deletes the classifier of the same name, if it exists. If this flag is set to false and a classifier with the same name exists, this action returns false. This parameter supports SmartParameters. The type of this parameter is string only to support SmartParameters. Internally, it is treated as a boolean. Specify '1' for true and '0' for false. Any other value is ignored and the action is default to false.

string appendToCSV - If an existing csv file found, this parameter indicates whether the training data from this batch should be appended to it or that it should be overwritten. If true, the classifier is trained with the data in the csv file and the new data from this batch. If false, the existing data in the csv file is overwritten and the classifier is trained with just the data in this batch. This parameter supports SmartParameters. The type of the parameter is string only to support SmartParameters. Internally, it is treated as a boolean. Specify '1' for true and '0' for false. Any other value is ignored and the action is default to true.

Returns

True, action is successful. Otherwise, False.

Level

Batch level.

Details

This action creates or replaces an NLC Classifier. Pages of the batch are used as the training data. It is expected that the batch is in a certain format. In particular, the batch needs to be divided into set documents where the 'Type' associated with the document is class that is created in NLC. All of pages in that document are treated as training data for that class. So, for example, if you wanted to train the classifier to recognize the main page and the trailing pages of a mortgage application, you would create a batch with two documents: Main_Page and Trailing_Page. Put all of the training page for the main page under the Main_Page document and the trailing pages under Trailing_Page document.

Example

```
NLCSetLanguage ("en")
NLCSetCredentials ("@APPVAR (values/gen/NLCUserName) ", "@APPVAR (values/adv/NLCPass
word) ")
NLCTrain ("@APPPATH (runtime)+\+..+\+CSV", "@APPVAR (values/gen/NLCClassifierName) "
, 0, 1)
```

Parent topic: [DocumentAnalytics.NaturalLanguageClassifier actions](#)

Documentum actions

Use the Documentum actions to upload documents to an EMC Documentum repository.

The Documentum Connector actions integrate Datacap applications with the Documentum repository. You run these actions to access the Documentum server, set up document attributes and folders on the server, and upload documents to the server for storage.

- [DM_Logon](#)
Creates the connection to the Documentum repository into which you can upload pages and documents.
- [DM_SetContentType](#)
Sets the content type to define in the repository for the object, for example TIFF, JPEG, DOC.
- [DM_SetFolderName](#)
Specifies the name of the Documentum folder where Datacap places the uploaded file on the Documentum system.
- [DM_SetObjectName](#)
Sets the name of the file that you are uploading as it appears in the Documentum repository.
- [DM_UploadDocument](#)
Uploads all of the pages in the document.
- [DM_UploadPage](#)
Uploads the selected page from the document.

Parent topic: [Global actions](#)

DM_Logon

Creates the connection to the Documentum repository into which you can upload pages and documents.

Syntax

```
bool DM_Logon (string sRepositoryDomain, string sRepositoryName, string sUserID,
string sPassword)
```

Parameters

string sRepositoryDomain

string sRepositoryName

string sUserID

String sPassword

Parameters

- sRepositoryDomain: The machine name for the repository.
- sRepositoryName: The name of the repository.
- sUserID: The userid for logon.
- sPassword: The password
-

The parameters cannot be blank. Smart parameters are supported.

Note: Use a Smart parameter to obtain the password from the application service instead of hardcoding it in the rules.

Returns

True, if the logon is successful. Otherwise, False.

Level

This action can be called at any level but is recommended to be called at the batch level. It must be called only once per task.

Details

Creates the connection to the repository where the pages are uploaded. This action must be called before DM_UploadPage or DM_UploadDocument. The user ID must have write permission or files cannot be uploaded.

It is recommended that you create an advanced value in the custom values tab in the Application Manager to store your password.

Example:

```
DM_Logon("machinename", "repository", "userid", "password")
```

```
DM_Logon("machinename", "repository", "userid",
"@APPVAR(values/adv/MyPassword) ")
```

This example uses the Smart parameter @APPVAR to obtain the password from the advanced value section of the application manager. The custom value name is "MyPassword".

Parent topic: [Documentum actions](#)

Related reference:

[DM_UploadPage](#)

[DM_UploadDocument](#)

DM_SetContentType

Sets the content type to define in the repository for the object, for example TIFF, JPEG, DOC.

Syntax

```
bool DM_SetContentType (StrParam)
```

Parameters

The repository defined type for this page. Smart parameters are supported.

Returns

Always True.

Level

Any level.

Details

Sets the type for the page to be uploaded. This type must be pre-defined in the repository. This action does not test that the specified type exists in the repository. If the specified content type is incorrect, the upload action will report the error.

This action must be called prior to `DM_UploadPage` or `DM_UploadDocument`.

Example:

```
DM_SetFolderName ("/folder1/folder2")
DM_SetContentType ("tiff")
DM_SetObjectName ("@ID")
DM_UploadDocument ()
```

Parent topic: [Documentum actions](#)

Related reference:

[DM_UploadPage](#)

[DM_UploadDocument](#)

DM_SetFolderName

Specifies the name of the Documentum folder where Datacap places the uploaded file on the Documentum system.

Syntax

```
bool DM_SetFolderName (StrParam)
```

Parameters

Repository folder where the file will be uploaded. Smart parameters are supported.

Returns

Always True.

Level

Any level.

Details

The name of the cabinet / folder that will hold the uploaded file. The path specification to the folder can be specified using a typical folder syntax separated by forward slashes, such as /folder/anotherfolder/finalfolder. Alternatively, the target folder can be specified by the object ID of the folder as defined in the repository, without any slashes.

This action does not confirm that the folder actually exists in the repository. If the specified content type is incorrect, the upload action will report the error. This action must be called prior to `DM_UploadPage` or `DM_UploadDocument`.

Example:

```
DM_SetFolderName("/folder1/folder2")
DM_SetContentType("tiff")
DM_SetObjectName("@ID")
DM_UploadPage()
```

This example shows the folder path as it exists in the repository.

```
DM_SetFolderName("0c0022538000252d")
DM_SetContentType("tiff")
DM_SetObjectName("@ID")
DM_UploadPage()
```

This example uses the object ID of the destination folder in the repository.

Parent topic: [Documentum actions](#)

Related reference:

[DM_UploadPage](#)

[DM_UploadDocument](#)

DM_SetObjectName

Sets the name of the file that you are uploading as it appears in the Documentum repository.

Syntax

```
bool DM_SetObjectName (StrParam)
```

Parameters

The name for the uploaded file as it will appear in repository. Smart parameters are supported.

Returns

Always True.

Level

Any level.

Details

This action is used to set the name of the uploaded file. This is not the file name as it exists in the batch, but the final name that will be used when viewed in the repository.

This action must be called prior to `DM_UploadPage` or `DM_UploadDocument`.

Example:

```
DM_SetFolderName("/folder1/folder2")
DM_SetContentType("tiff")
DM_SetObjectName("@ID")
DM_UploadDocument()
```

Parent topic: [Documentum actions](#)

Related reference:

[DM_UploadPage](#)

[DM_UploadDocument](#)

DM_UploadDocument

Uploads all of the pages in the document.

Syntax

```
bool DM_UploadDocument ()
```

Parameters

None.

Returns:

True, if all of the pages within the document are uploaded to the repository. Otherwise, False.

If any of the files for the document are missing from the batch, it is not considered an error and the batch will not abort. If the upload to Documentum fails due to a different reason, the document upload will stop and the batch will be set to abort.

Level

Document level.

Details

This action will upload all of the pages that are attached to a document. An XML file called `DM_Uploaded.xml` is created in the batch directory. This file lists all of pages that have been uploaded.

`DM_Logon` must have been previously called. Additionally, the destination folder, final object name and content type must have been previously set. When uploading a document using this action, all pages must be of the

same type.

Example:

```
DM_SetFolderName ("/folder1/folder2")
DM_SetContentType ("tiff")
DM_SetObjectName ("@ID")
DM_UploadDocument ()
```

Parent topic: [Documentum actions](#)

Related reference:

[DM_SetFolderName](#)

[DM_SetContentType](#)

[DM_SetObjectName](#)

DM_UploadPage

Uploads the selected page from the document.

Syntax

```
bool DM_UploadPage ()
```

Parameters

None.

Returns

True, if the page is uploaded to the repository. Otherwise, False.

If the file fails to upload or if the file is missing from the batch, the batch will be set to abort.

Level

Page level.

Details

This action uploads the current page to the repository. An XML file called DM_Uploaded.xml is created in the batch directory. This file lists all of pages that have been uploaded.

DM_Logon must have been previously called. Additionally, the destination folder, final object name and content type must have been previously set.

Example:

```
DM_SetFolderName ("/folder1/folder2")
DM_SetContentType ("tiff")
DM_SetObjectName ("@ID")
DM_UploadPage ()
```

Parent topic: [Documentum actions](#)

Related reference:

[DM_SetFolderName](#)

[DM_SetContentType](#)

Email actions

Use the email actions to compose and then send an email by using CDOSYS and an SMTP server. These actions also support Outlook, which requires the Outlook user to be logged on to the computer and security prompts might be displayed for each message.

Important: If any concurrently running threads might be using the same sending email address, your Email-related tasks cannot be run in a multi-threaded configuration. Instead, use single-threaded tasks only. For information about configuring threads in Rulerunner, see [Rulerunner thread configuration](#). For information about the sending email address, see [SetSender](#).

The Email actions specify path and file name for attachments, email recipients, and email subject.

- [SendEMail](#)
Sends an email.
- [SetAttachment](#)
Adds a file attachment to an email.
- [SetBlindCarbonCopyRcpts](#)
Sets Blind Carbon Copy recipients' email address(es).
- [SetCarbonCopyRcpts](#)
Set Carbon Copy recipients' email address(es).
- [SetEmailBody](#)
Sets the text of the email's body.
- [SetMailServer](#)
Configures the mail server to use for sending mail. Use this action only if you are sending emails with CDOSYS.
- [SetRecipients](#)
Sets email recipients' address(es).
- [SetSender](#)
Sets the sending email address.
- [SetSubject](#)
Sets the text for the email's Subject field.

Parent topic: [Global actions](#)

SendEMail

Sends an email.

Syntax

```
bool SendEMail ()
```

Parameters

None.

Returns

False, if the rule does not include a previous SetRecipients action, or if the email cannot be sent. Otherwise, True. If the email cannot be sent, the batch is set to abort.

Level

All levels.

Details

Sends an email that is assembled by previous actions. Typically, this is the final action in an email rule set. At a minimum, the `SetSender` and `SetRecipients` actions must be called before sending an email.

After the email is sent, this action will discard the contents of the email in memory. Calls to the email actions after `SendEmail` causes the creation of a new email message.

Example:

```
SetSender("paul@adomain.com")
SetRecipients("lisa@adomain.com,beth@adomain.com")
SetSubject("Document Integrity")
SetEMailBody("Document Page Types and counts are accurate. Thanks for your
help.")
SendEMail()
```

Parent topic: [Email actions](#)

SetAttachment

Adds a file attachment to an email.

Syntax

```
bool SetAttachment (StrParamMW)
```

Parameters

The file's path, name and extension. Smart Parameters are supported.

Returns

False if the file does not exist or cannot be attached. Otherwise, True.

Level

All levels.

Details

Attaches the specified file to the current email.

Example:

```
SetAttachment("h:\MyDir\MQSW\export\+@BATCHID+.txt")
```

This example attaches the Export file of the current batch to the email.

Parent topic: [Email actions](#)

SetBlindCarbonCopyRcpts

Sets Blind Carbon Copy recipients' email address(es).

Syntax

```
bool SetBlindCarbonCopyRcpts (StrParam)
```

Parameters

The email addresses to receive a copy of the email as a blind carbon copy. You can enter multiple email addresses separated by commas.

Returns

False if you do not enter an email addresses parameter, if the address is rejected by the mail system or if the email object cannot be initialized. Otherwise, True.

Invalid email addresses may not be reported until SendEMail is called.

Level

All levels.

Details

Adds addresses to the Bcc (Blind Carbon Copy) portion of the email's header.

Example:

```
SetRecipients("lisa@monarchy.com")  
SetBlindCarbonCopyRcpts("james@regency.com")
```

Parent topic: [Email actions](#)

Related reference:

[SetCarbonCopyRcpts](#)

[SetRecipients](#)

SetCarbonCopyRcpts

Set Carbon Copy recipients' email address(es).

Syntax

```
bool SetCarbonCopyRcpts (StrParam)
```

Parameters

Email addresses to receive a copy of the email as a carbon copy. You can enter multiple email addresses separated by commas.

Returns

False if you do not enter an email addresses parameter, if the address is rejected by the mail system or if the email object cannot be initialized. Otherwise, True.

Invalid email addresses may not be reported until SendEMail is called.

Level

All levels.

Details

Adds addresses to the Cc (Carbon Copy) portion of the email's header.

Example:

```
SetRecipients("lisa@adomain.com")
SetCarbonCopyRcpts("cindy@anotherdomain.org")
```

Parent topic: [Email actions](#)

Related reference:

[SetBlindCarbonCopyRcpts](#)

[SetRecipients](#)

SetEmailBody

Sets the text of the email's body.

Syntax

```
bool SetEmailBody (StrParamMW)
```

Parameters

The email message text. Smart Parameters are supported.

Returns

False if the mail object cannot be initialized. Otherwise, True.

Level

All levels.

Details

Sets the text of the email's body.

Example:

```
SetSubject("Document Integrity")
SetEmailBody("Document Page Types and counts are accurate. Thanks for your
help.")
```

Parent topic: [Email actions](#)

Related reference:

SetMailServer

Configures the mail server to use for sending mail. Use this action only if you are sending emails with CDOSYS.

Syntax

```
bool SetMailServer (StrParam)
```

Parameters

The IP or DNS address of the outgoing mail (SMTP) server.

Returns

Always True.

Level

All levels.

Details

Sets the address of the outgoing mail (SMTP) server. This server might be the same mail server that you configure in your mail program. The server must be accessible from the computer that runs the email actions. This action must be the first action in an email rule, if the CDOSYS object is being used.

Use this action only if you are sending emails with CDOSYS. To use CDOSYS, this action must be called before any of the other email actions. If this action is not called before other email actions, these actions use Outlook for sending emails.

You can use Email actions to direct a task to compose and send emails that contain information and attachments. Email actions use the Windows CDOSYS library to send email by your preferred SMTP mail server. The CDOSYS object is included with Windows 2000 and higher versions. Alternatively, Email actions can use the Outlook object but it is not recommended.

One of these two libraries (CDOSYS or Outlook) must be registered on the computer that runs the rules that employ email actions.

Outlook is primarily useful for demonstration purposes because it is not suitable for unattended operation. It requires the Outlook user to be logged in to the computer, and security prompts might be displayed for each message sent.

Example:

```
SetMailServer ("mail.YourISP.com")
SetSender ("paul@adomain.com")
SetRecipients ("lisa@adomain.com")
SetSubject ("Document Integrity")
SetEMailBody ("Document Page Types and counts are accurate. Thanks for your
help.")
SendEMail ()
```

Parent topic: [Email actions](#)

SetRecipients

Sets email recipients' address(es).

Syntax

```
bool SetRecipients (StrParam)
```

Parameters

Email address(es) of recipient(s). You can either call this action multiple times to add multiple recipients, or you can enter multiple email addresses separated by commas.

Returns

False if you do not enter an email addresses parameter, if the address is rejected by the mail system or if the email object cannot be initialized. Otherwise, True.

Invalid email addresses may not be reported until SendEMail is called.

Level

All levels.

Details

The email address of the email's primary recipients.

Example:

```
SetRecipients ("lisa@adomain.com, Joe@adomain.com")
```

Parent topic: [Email actions](#)

Related reference:

[SetCarbonCopyRcpts](#)

[SetBlindCarbonCopyRcpts](#)

SetSender

Sets the sending email address.

Syntax

```
bool SetSender (StrParam)
```

Parameters

The sender's email address.

Returns

False if the mail object cannot be initialized. Otherwise, False.

Invalid email addresses may not be reported until SendEMail is called.

Level

All levels.

Details

Sets the email address of the sender for the current email. When using the CDOSYS object, use this action. When using the Outlook object, the current email account is used as the sender.

Example:

```
SetRecipients("lisa@adomain.com")
SetSender("paul@adomain.com")
```

Parent topic: [Email actions](#)

SetSubject

Sets the text for the email's Subject field.

Syntax

```
bool SetSubject (StrParamMW)
```

Parameters

The subject line of the email. Smart Parameters are supported.

Returns

False if the mail object cannot be initialized. Otherwise, True.

Level

All levels.

Details

Sets the text for the email Subject field. It is recommended that the subject line be no longer than 78 characters as this is a common subject line length limitation. Some systems may support even shorter lengths, truncating the subject. Our testing has been successful with lengths up to 255 characters. It is recommended to test your settings and use lengths appropriate for your systems.

Example:

```
SetSubject("Document Integrity")
```

Parent topic: [Email actions](#)

Equalize actions

Use the Equalize action to equalize the x and y resolutions of an image.

The Equalize actions convert an image with different x and y resolutions to one with the same x and y resolutions.

- [EqualizeUnbalancedImage](#)
Resolves differences in the dpi (dots per inch) resolutions along the horizontal (X) and vertical (Y) planes of one or more faxed images in a batch.

Parent topic: [Global actions](#)

EqualizeUnbalancedImage

Resolves differences in the dpi (dots per inch) resolutions along the horizontal (X) and vertical (Y) planes of one or more faxed images in a batch.

Syntax

```
bool EqualizeUnbalancedImage (StrParam)
```

Parameters

Integer value which determines the cut-off point for the resolution which should be equalized:

EqualizeUnbalancedImage(0), for example, specifies that there is no cut-off point: all images will be subject to equalization.

EqualizeUnbalancedImage(20) establishes a cut-off point of 200(x)/180(y). This means that the action will equalize all images with this resolution ratio and more (200/180, 200/160 etc.) - but will ignore all images with balance ratios less than 200/180 (in this example.)

Attention: Standard Mode fax resolution in Dots per Inch (DPI) is 204/98; Fine Mode fax resolution is 204/196.

Returns

False if the parameter is not numeric or if the rule containing the action is not bound to a Page object of the Document Hierarchy. Otherwise, True.

Level

Page level only.

Details

Resolves differences in the dpi (dots per inch) resolutions along the horizontal (X) and vertical (Y) planes of one or more faxed images in a batch.

The action selects images from the batch in response to the parameter you enter, and produces new images with 200 x 200 dpi.

Example

```
EqualizeUnbalanceImage (0)
```

Parent topic: [Equalize actions](#)

Ewsmail actions

Use the Ewsmail actions to import image file attachments from an Exchange Server into the current batch by using Exchange Web Service (EWS).

Important: If any concurrently running threads might be using the same mail account, your Ewsmail-related tasks cannot be run in a multi-threaded configuration. Instead, use single-threaded tasks only. For information about configuring threads in Rulerunner, see [Rulerunner thread configuration](#). For information about the mail account, see [ex_login](#).

The `ex_scan` action polls the Exchange Server and imports image attachments until the batch reaches the specified size (`ex_max_docs`) or until the wait time (`ex_wait_time`) expires.

- [ex_abort_time](#)
Specifies how long to wait before you stop running the current batch if, for example, the Exchange Server is unavailable.
- [ex_done_folder](#)
Specifies the mailbox subfolder to which email messages are moved after the attachment was imported.
- [ex_EMLOption](#)
Creates a one page document per email containing an .eml file.
- [ex_ews_version](#)
Select the Exchange Server version.
- [ex_HTTP_timeout](#)
Specifies the maximum time to wait for an HTTP request or response from Exchange Server.
- [ex_load_properties_option](#)
Option to flag .
- [ex_login](#)
Specifies the Exchange Server and mail account
- [ex_logout](#)
Disconnect from the mail server
- [ex_max_docs](#)
Specifies maximum number of emails to include in a single batch.
- [ex_problem_folder](#)
Specifies folder for problem emails
- [ex_scan](#)
Poll the specified mail server for incoming email with image attachments
- [ex_types](#)
Specifies valid image attachment extensions
- [ex_wait_time](#)
Specifies the maximum time to wait for input emails for a single batch.

Parent topic: [Global actions](#)

ex_abort_time

Specifies how long to wait before you stop running the current batch if, for example, the Exchange Server is unavailable.

Member of namespace

Ewsmail

Syntax

```
bool ex_abort_time (int nSecs)
```

Parameters

nSecs
Type: int

Parameters

nSecs : The number of seconds to wait.

Returns

Always True.

Level

Batch level.

Details

The action will wait the specified time before returning when an abort occurs. This action can be useful to prevent a large number of aborted batches due to an abort condition. For example, if the email server should become unavailable for a time, the abort timeout will limit the amount of aborted batches until the mail server becomes available again.

If this action is not called, the default value of 30 seconds will be used.

Example:

```
ex_wait_time("20")  
ex_abort_time("60")  
ex_max_docs("200")  
ex_scan
```

Parent topic: [Ewsmail actions](#)

Related reference:

[ex_scan](#)

ex_done_folder

Specifies the mailbox subfolder to which email messages are moved after the attachment was imported.

Member of namespace

Ewsmail

Syntax

```
bool ex_done_folder (string folder)
```

Parameters

folder
Type: string
Destination folder for successfully imported emails

Parameters

folder : Destination folder for successfully imported emails

Returns

Always True.

Level

Batch level.

Details

Specifies the name of the email folder into which successfully imported emails are placed. This folder must be a subfolder of the email account's Inbox. When an email is processed and the attachment is imported, the email is moved to the folder name specified by this action.

If this action is not called, the default value of Done is used.

Example:

```
ex_done_folder("Imported")  
ex_problem_folder("Failed")
```

Parent topic: [Ewsmail actions](#)

Related reference:

[ex_scan](#)

ex_EMLOption

Creates a one page document per email containing an .eml file.

Member of namespace

Ewsmail

Syntax

```
bool ex_EMLOption (int folder)
```

Parameters

folder
Type: int
Optional - use a nonzero value to store one .eml file per email.

Parameters

folder : Optional - use a nonzero value to store one .eml file per email.

Returns

Always True.

Level

Batch level.

Details

If set, the `ex_scan` action creates a one page document containing the email and attachments in an .eml file; no attachment pages are created. When called with a non-zero parameter, the `ex_scan` function does not create pages for each attachment, instead one page is created per email document, containing an .eml file suitable for subsequent processing using eDocument Conversion actions.

When you use this action, pages for attachments are not created, and variables for those attachments are not set.

Example:

```
ex_EMLOption(1)
ex_scan()
```

Parent topic: [Ewsmail actions](#)

Related reference:

[ex_scan](#)

ex_ews_version

Select the Exchange Server version.

Member of namespace

Ewsmail

Syntax

```
bool ex_ews_version (int version)
```

Parameters

version

Type: int

Parameters

The value 0, 1 or 2, indicating the following:

- 1 = Exchange 2007 SP1
- 2 = Exchange 2010
- 0 = latest version (default)

Returns

Always True.

Level

Batch level, Open event.

Details

Each version of Exchange uses a slightly different communication protocol. Use this action to set the expected version.

In order to connect successfully with:

- Exchange 2007 SP1, call this action with parameter 1 before `im_login`.
- Exchange 2010, call action with parameter 2 before `im_login`.
- The latest version known by the .NET library in use, which is .NET 3.5 and is currently Exchange 2010. This action must be called with parameter 0.

If this action is not called at all it uses the default 0, the latest version.

Example:

```
ex_ews_version(1)
ex_login("myemailserver/Exchange.asmx", "Username@Org", "password")
```

Parent topic: [Ewsmail actions](#)

Related reference:

[ex_login](#)

ex_HTTP_timeout

Specifies the maximum time to wait for an HTTP request or response from Exchange Server.

Member of namespace

Ewsmail

Syntax

```
bool ex_HTTP_timeout (int nSecs)
```

Parameters

String nSecs

Parameters

nSecs : The maximum number of seconds to wait.

Returns

Always True.

Level

Batch level Open event only.

Details

The maximum time to wait for an HTTP request or response from Exchange Server. For example, this may be used by the `ex_scan` action while trying to ingest a very large email that may otherwise result in import failing due to the operation timing out. This timeout is in addition to any server-side timeout configuration settings.

If this action is not called, the timeout defaults to 100 seconds.

Example:

```
ex_HTTP_timeout("60")
ex_scan()
```

This example causes the timeout to be set to 60 seconds. When ingesting emails, any response exceeding this value may fail.

Parent topic: [Ewsmail actions](#)

Related reference:

[ex_login](#)

[ex_scan](#)

ex_load_properties_option

Option to flag .

Member of namespace

Ewsmail

Syntax

```
bool ex_load_properties_option (int nOption)
```

Parameters

nOption

Type: int

Parameters

nOption : 0 to load base properties, 1 to load limited extended base properties, 2 to load extended properties

Returns

Always True.

Level

Batch level Open event only.

Details

Used to optionally load certain properties at once. If this action is not called, the default of 0 will be used to load base properties.

Example:

```
ex_load_properties_option("1")
ex_scan()
```

This example reduces the number of times you have to access the Exchange Server because the additional properties are loaded once.

Parent topic: [Ewsmail actions](#)

Related reference:

[ex_scan](#)

ex_login

Specifies the Exchange Server and mail account

Member of namespace

Ewsmail

Syntax

```
bool ex_login (string hostname, string username, string password)
```

Parameters

hostname

Type: string

URL of Exchange Web Service, ends with /Exchange.asmx (Smart parameters are supported)

username

Type: string

Username@Org for mail account and organization (blank to use Windows Authentication, Smart parameters are supported)

password

Type: string

Password for mail account (blank if none or Windows Authentication, Smart parameters are supported)

Parameters

- hostname : URL of Exchange Web Service, ends with /Exchange.asmx
- username : Username@Org for mail account and organization (blank to use Windows Authentication)
- password : Password for mail account (blank if none or Windows Authentication)

All parameters support smart parameters.

Returns

True if the login succeeds. Otherwise, False.

Level

Batch level, open event.

Details

Connects to the mail server using the specified account information. Login credentials are for a Microsoft Exchange mail server.

The mail account must contain an Inbox folder and separate folders for messages imported (Done) and errors (Problem). The Done and Problem folders must be subfolders of the email account's Inbox. The names of these folders can be specified using the `ex_done_folder` and `ex_problem_folder` actions.

The Microsoft Exchange Email actions are designed to scan an email Inbox for incoming mail messages, and place selected messages into a new batch. It is possible to ignore all messages except those containing specific attachment types. The actions are typically assigned to a Task that is executed by an unattended Rulerunner station. Multiple Inboxes can be scanned by stringing together a set of login/scan/logout actions for each Inbox, or by assigning an Inbox to each input task.

Example:

```
ex_login("mymailserver/Exchange.asmx", "Username@Org", "password")
```

Parent topic: [Ewsmail actions](#)

Related reference:

[ex_logout](#)

ex_logout

Disconnect from the mail server

Member of namespace

Ewsmail

Syntax

```
bool ex_logout ()
```

Parameters

None.

Returns

Always True.

Level

Batch level Open or Close event only.

Details

Closes the connection to the mail server. Call once for each batch after `ex_login` and `ex_scan` have completed.

Example:

```
ex_logout()
```

Parent topic: [Ewsmail actions](#)

ex_max_docs

Specifies maximum number of emails to include in a single batch.

Member of namespace

Ewsmail

Syntax

```
bool ex_max_docs (int nDocs)
```

Parameters

nDocs

Type: int

The maximum number of emails in a batch.

Parameters

nDocs : The maximum number of emails in a batch.

Returns

Always True.

Level

Batch level Open event only.

Details

The import of emails into the batch will stop when this email limit is reached or when the maximum wait time has been reached. While waiting for new mail to arrive, the configured mailbox will be polled every two seconds to check for waiting mail.

If this action is not called, the default value of 100 is used. The actual amount of emails included in the batch could be less than this maximum.

Note: The value indicates the number of emails with expected attachments that will be processed. If a waiting email contains an unexpected attachment, it is not counted against the total maximum count.

It is also possible for an email to contain more than one valid attachment type. When this happens, it is possible for the number of items input to the batch can be greater than the number of configured emails. For example, if `ex_max_docs` is set to 5 and if there are 5 waiting emails and each email has 2 attachments, the total number of attachments that are included in the batch is 10.

Example:

```
ex_max_docs("50")
ex_scan()
```

This example causes the scan operation to limit the number of emails in a batch to a maximum of 50.

Parent topic: [Ewsmail actions](#)

Related reference:

[ex_scan](#)

ex_problem_folder

Specifies folder for problem emails

Member of namespace

Ewsmail

Syntax

```
bool ex_problem_folder (string folder)
```

Parameters

folder

Type: string

Destination folder for problem email

Parameters

folder : Destination folder for problem email

Returns

Always True.

Level

Batch level.

Details

When an email is processed and the attachment is not one of the expected types, the email is moved to the folder name specified by this action.

This folder must be a subfolder of the email account's Inbox.

If this action is not called, the default value of Problem is used.

Example:

```
ex_done_folder("Imported")
ex_problem_folder("Failed")
```

Parent topic: [Ewsmail actions](#)

Related reference:

[ex_scan](#)

ex_scan

Poll the specified mail server for incoming email with image attachments

Member of namespace

Ewsmail

Syntax

```
bool ex_scan ()
```

Parameters

None.

Returns

False if the operation fails, and the action will also pause before returning based on the configured abort time configured by `ex_abort_time`. Otherwise, True.

If no selected emails were available, the action returns True and also pauses before returning based on the wait time configured using `ex_wait_time`.

Level

Batch level Open event only.

Details

Scans emails in the Inbox for specified attachments, imports selected emails with attachments into the batch. Call once for each batch. A connection to the email server must have previously been established using the `ex_login` action.

Each input email (document) will contain the following variables set in the document hierarchy:

- `TYPE` : Always set to Document.
- `MessageID` : The email message ID.
- `Subject` : The email subject.
- `Body` : The text within the email.
- `DateSent` : The sent date stamp on the email.
- `From` : The email sender.
- `To` : The email recipients.
- `Priority` : The state of the email importance flag.
- `DateReceived` : The received date stamp on the email.

Each email attachment (page) will have the following variables set:

- `TYPE` : Always set to "Other".
- `IMAGEFILE` : The name of the attachment as saved on disk.

- ATTACHNAME : The name of the attachment.

The following batch level variable is created:

- EmailCount : The number of emails scanned into the batch.

Note: If the configured Done or Problem folders do not exist, the email will be moved to the Deleted Items folder.

Waiting emails are processed in blocks. The amount of emails processed in a block will be the number of waiting emails, or the number of emails remaining from the value set in `ex_max_docs`, up to a maximum of 50. After a block of emails are processed, the wait time is checked. If the wait time has not been reached and the maximum number of emails has not been reached, then another block of emails are processed. If the wait time has been reached, then no more emails are input to the batch.

It is recommended that the subject line be no longer than 78 characters as this is a common subject line length limitation. Some systems may support even shorter lengths, truncating the subject. Our testing has been successful with lengths up to 255 characters. It is recommended to test your settings and use lengths appropriate for your systems.

Use `ex_max_docs` to configure the number of emails processed per batch. The number of items included in the batch can be different than this amount, see `ex_max_docs` for more details.

Example:

```
ex_wait_time("20")
ex_abort_time("40")
ex_max_docs("50")
ex_types("jpg,tif")
ex_scan()
```

Parent topic: [Ewsmail actions](#)

Related reference:

[ex_login](#)

[ex_wait_time](#)

[ex_max_docs](#)

ex_types

Specifies valid image attachment extensions

Member of namespace

Ewsmail

Syntax

```
bool ex_types (string extensions)
```

Parameters

extensions

Type: string

Comma-separated list of file image file attachment extensions to import, with or without period. If a blank extension is supplied, `ex_scan` will import messages with no attachments.

Parameters

extensions : A comma-separated list of file image file attachment extensions to import, with or without period. If a blank extension is supplied, ex_scan will import messages with no attachments.

Returns

Always True.

Level

Batch level.

Details

This action specifies the allowable email attachment types.

If this action is not called, the default value of .pdf is used.

If this action is called and no attachment types are specified, all emails and any attachments are added to the batch.

If attachment types are specified and the email contains:

- Only the specified attachment types, the email and attachments are added to the batch
- Only unspecified attachment types, or if it contains a mix of specified and unspecified attachment types, the email is moved to the Problem folder
- No attachments, the email is moved to the Problem folder

Example:

```
ex_wait_time("20")
ex_abort_time("40")
ex_max_docs("200")
ex_types("jpg,tif")
```

Parent topic: [Ewsmail actions](#)

ex_wait_time

Specifies the maximum time to wait for input emails for a single batch.

Member of namespace

Ewsmail

Syntax

```
bool ex_wait_time (int nSecs)
```

Parameters

nSecs

Type: int

Parameters

nSecs : The maximum number of seconds to wait.

Returns

Always True.

Level

Batch level Open event only.

Details:

The maximum time to wait for input files for a single batch. Used by the `ex_scan` action after the first email has been processed to determine how long to wait for the batch to fill up.

The import of emails into the batch will stop when the wait limit is reached or when the maximum emails per batch has been reached. While waiting for new mail to arrive, the mailbox will be polled every two seconds to check for waiting mail. The action will continue to include new emails into the batch until this wait time is reached or the maximum number of emails per batch is reached.

If this action is not called, the default wait time of 5 seconds will be used.

Example:

```
ex_wait_time("20")
ex_scan()
```

This example causes the scan operation to wait for 20 seconds for additional email, after processing the first email, before finishing the scan operation.

Parent topic: [Ewsmail actions](#)

Related reference:

[ex_scan](#)

Export actions

Use the Export actions to set up and write information to the export text file.

The Export actions can setup lineitem values, batch, document and page variables, path locations, before exporting the information.

- [BatchVariable_ExportValue](#)
Exports the value contained in the specified batch-level variable.
- [BlankFields](#)
Inserts n blank fields into the Export file, adjacent to the current field.
- [BlankLines](#)
Inserts n blank lines into the Export file.
- [BPilot](#)
Exports the value assigned to the Batch Pilot property designated as the parameter.
- [CloseExportFile](#)
Closes the currently opened Export file.

- [DCOProperty](#)
Exports the value assigned to the DCO property that you designate as the parameter.
- [DocumentVariable_ExportValue](#)
Exports the value that is contained in the specified Document-level variable.
- [ExportAllFields](#)
Exports all field values on the current page, including values of Line Item Detail sub-fields - with exceptions.
- [ExportFieldValue](#)
Exports the specified Field object's value to the Export file.
- [ExportMYValue](#)
Exports the current field value to the Export file.
- [ExportSmartParameter](#)
Exports an evaluated smart parameter value to the Export file.
- [ExportToBatchDir](#)
Specifies that the path for the current Export file (txt) is the Batch directory.
- [Filler](#)
Adds a string of identical filler characters to the Export.
- [FixedLenLJ](#)
Exports a specified number of characters from a field's left end (left-justified).
- [FixedLenRJ](#)
Exports a specified number of characters from a field's right end (right-justified).
- [GetDATE](#)
Exports today's Date in the format specified as the parameter.
- [GetProfileString](#)
Accesses a Settings file (.ini) and adds a value in that file to your Export file.
- [GetTime](#)
Exports the current Time in the format specified as the parameter.
- [LineItem_AddElement](#)
Includes the specified Line Item Field object as an element of a Line Item Array.
- [LineItem_BlankFields](#)
Includes the specified number of blank fields as elements of a Line Item Array.
- [LineItem_ClearElements](#)
Clears values in the Line Item Array.
- [LineItem_ExportElements](#)
Exports the captured values in a page's Line Item Array that have been populated with LineItem_AddElement actions.
- [LineItem_SmartParameter](#)
Add a smart parameter algorithm as an element of a Line Item Array.
- [NewLine](#)
Starts a new line in your Export file.
- [PageVariable_ExportValue](#)
Exports runtime values assigned to a variable of the bound Page object of the Document Hierarchy.
- [ResetFieldVariables](#)
Resets the variables of the bound Field object of the Document Hierarchy.
- [SaveFilePathAsVariable](#)
Saves the path and name of your Export file to the variable specified by the parameter.
- [SetCSV](#)
Ensures that all exported values are delimited by a comma separator.
- [SetElementSeparator](#)
Ensures that all exported values are delimited by a separator designated as the parameter.
- [SetExportFileEncodingAsASCII](#)
Determines if the Export file is created with ASCII or UNICODE encoding.

- [SetExportPath](#)
Specifies the path to the Export file's location. Alternatively, you can use a Smart Parameter to identify a Paths.ini file that has a set of path parameters for your application.
- [SetExtensionName](#)
Assigns an extension to the current Export file.
- [SetFileName](#)
Assigns a name to the current Export file.
- [SetFill](#)
Sets the filler character to be used to expand the current value of a field in a flat file, if the field's allowable length is greater than the length of its current export value.
- [SetFixedLength](#)
Uses the Numeric value you enter as a parameter to establish a fixed length of a value exported from the current field.
- [SetIgnoreFieldStatus](#)
Assigns a Numeric value to the application's SetIgnoreStatus variable. Any field with this status cannot export data to an Export file or database.
- [SetJustified](#)
Right-justifies or left-justifies a field's exported values.
- [SetOMR_Separator](#)
For multi-punch OMR fields, uses the parameter's value as the separator character.
- [SetSpaceFill](#)
Specifies the use of the ASCII 32 space as the global filler value to be used to expand the current value of a field in a flat file, if the field's allowable length is greater than the length of its current export value.
- [SetZeroFill](#)
Sets the ASCII 48 zero as the global filler value to be used to expand the current value of a field in a flat file, if the field's allowable length is greater than the length of its current export value.
- [Text](#)
Places a string into the Export file.
- [Variable_ExportValue](#)
Exports the value assigned to a variable of the current object of the Document Hierarchy.
- [Variable_IsValue](#)
Checks to see if the parameter value matches the value assigned to a variable of the current object of the Document Hierarchy.

Parent topic: [Global actions](#)

BatchVariable_ExportValue

Exports the value contained in the specified batch-level variable.

Syntax

```
bool BatchVariable_ExportValue (StrParam)
```

Parameters

The name of the Batch variable.

Returns

Always True.

Level

Any level.

Details

Exports the value contained in the specified batch-level variable.

Example

```
BatchVariable_ExportValue("ED")
```

This action will export the value located in the ED Batch variable to your Export file.

Parent topic: [Export actions](#)

BlankFields

Inserts n blank fields into the Export file, adjacent to the current field.

Syntax

```
bool BlankFields (StrParam)
```

Parameters

A number indicating how many blank fields to add to the Export file.

Returns

Always True.

Level

All levels.

Details

The BlankFields action adds the number of fields you specify. The fields are blank; other actions direct the Export task to fill the fields.

Attention: Make sure you call SetCSV, and optionally SetElementSeparator, to set the separator values as desired for your export file. If either of these actions are not called before the BlankFields action, blank fields cannot be exported into the file because the default separator is set to no separator.

Example

```
SetCSV("TRUE")  
BlankFields("12")
```

Parent topic: [Export actions](#)

Related reference:

[SetCSV](#)

[SetElementSeparator](#)

BlankLines

Inserts n blank lines into the Export file.

Syntax

```
bool BlankLines (StrParam)
```

Parameters

A number n that indicates how many blank lines to add after the current line.

Returns

Always True.

Level

All level.

Details

Inserts n blank lines into the Export file.

Example

```
BlankLines ("4")
```

This action inserts four empty lines, leaving the insertion point for the next output on the following line. Additional output begins on the fifth line.

Parent topic: [Export actions](#)

Related reference:

[NewLine](#)

BPilot

Exports the value assigned to the Batch Pilot property designated as the parameter.

Syntax

```
bool BPilot (StrParam)
```

Parameters

The name of the Batch Pilot Property whose value is to be included in the Export file.

- BatchDir: The name and location of the application's Batches directory.
- BatchID: The Batch Number of the current batch (20020072.003, for example.)
- JobName: The name of the current User Application job (Main, for example.)
- Operator: The User ID of the operator currently processing the batch.
- PagesInBatch: A count of all pages in the batch.
- DocsInBatch: A count of all documents in the batch. Remember: in most configurations, a Recognition task reorganizes a batch into a series of documents and their pages.
- Priority: The processing priority assigned to the current batch ("10" = Low, "1" = High, "5" = Default). A task selects batches from its queue first according to Priority.

- **Station:** The Station ID of the workstation currently processing the batch.
- **TaskName:** The name of the task with the batch in its queue.
- **XtraBatchFieldValue:** The value in a custom field you've added to the Job Monitor's Batch Information Table.

Returns

False if the parameter is not a Batch Pilot property. Otherwise, True.

Level

Any level.

Details

Exports the value assigned to the Batch Pilot property that designated as the parameter.

Example

```
NewLine ()
Text ("BatchID:")
BPilot ("BatchID"
```

This sequence adds "BatchID:" followed by the current Batch ID into the Export file. For example:
BatchID: 20050019.001

Parent topic: [Export actions](#)

CloseExportFile

Closes the currently opened Export file.

Syntax

```
bool CloseExportFile ()
```

Parameters

None.

Returns

Always True.

Level

Any level.

Details

Closes the currently opened Export file.

This action usually belongs to its own RuleSet (ExportClose, for example), and applies to the Batch object of the Document Hierarchy. However, it can be used at any level.

Example

```
CloseExportFile()
```

Parent topic: [Export actions](#)

Related reference:

[SetFileName](#)

DCOProperty

Exports the value assigned to the DCO property that you designate as the parameter.

Syntax

```
bool DCOProperty (StrParam)
```

Parameters

The name of the DCO Property whose value is to be included in the Export file.

- **ID:** The value of an object's ID property. For a Batch object, this might be 20020072.003. You can apply this action at any level(s).
- **ImageName:** The name and location of a Page object's Image file: for example, c:\ParentDirectory\Invoices\batches\20030145.001\TM000001.tif.
- **Status:** The value assigned to an object's Status property. You can apply this action at any level(s).
- **Type:** The value assigned to an object's Type property. You can apply this action at any level(s).

Returns

False if the parameter is not a valid DCO Property. Otherwise, True.

Level

All levels.

Details

Exports the value assigned to the DCO Property that you designate as the parameter.

Example

```
NewLine()  
Text("Document: ")  
DCOProperty("ID")
```

If this sequence is applied to a Document object, the Export file for document 01 in batch 20050219.057 will look like: Document: 20050219.057.01.

Parent topic: [Export actions](#)

Related reference:

[BPilot](#)

DocumentVariable_ExportValue

Exports the value that is contained in the specified Document-level variable.

Syntax

```
bool DocumentVariable_ExportValue (StrParam)
```

Parameters

Document Variable Name.

Returns

Always True.

Level

Document, Page or Field level.

Details

Exports the value that is contained in the specified Document-level variable.

Example

A number of techniques can add variables to a Document object of the Document Hierarchy. These variables are listed as properties of the object in the Document Hierarchy Setup window.

The following example reads the value that is assigned to the *PageCount Document* variable and places it in the Export file.

```
DocumentVariable_ExportValue ("PageCount")
```

Parent topic: [Export actions](#)

ExportAllFields

Exports all field values on the current page, including values of Line Item Detail sub-fields - with exceptions.

Syntax

```
bool ExportAllFields ()
```

Parameters

None.

Returns

False, if the action is not used at the Page level. Otherwise, True.

Level

Page level.

Details

This action exports all field values of the current page. However, the action does not export those values of Field objects of the Document Hierarchy with:

1. A setup *NOEXPORT* variable of "1", or
2. A value of a runtime SetIgnoreStatus, a field status equal to the Numeric field status defined by a previously-run SetIgnoreFieldStatus action. For details, see the description of that action.

Example

```
ExportAllFields ()
```

Parent topic: [Export actions](#)

ExportFieldValue

Exports the specified Field object's value to the Export file.

Syntax

```
bool ExportFieldValue (StrParam)
```

Parameters

The name of the Field object whose value you want to export.

Returns

False if the parameter is not a Field object's name. Otherwise, True.

Level

Page level.

Details

Exports the specified Field object's value to the Export file. Will only export the last Field if multiple fields of the same field type are found.

Example

```
ExportFieldValue ("Date")  
ExportFieldValue ("Number")  
ExportFieldValue ("Total")
```

This sequence exports the current values stored in the Date, Number and Total fields.

Parent topic: [Export actions](#)

ExportMYValue

Exports the current field value to the Export file.

Syntax

```
bool ExportMYValue ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Exports the current field value to the Export file.

Example

```
ExportMYValue ()
```

Parent topic: [Export actions](#)

ExportSmartParameter

Exports an evaluated smart parameter value to the Export file.

Syntax

```
bool ExportSmartParameter (StrParam)
```

Parameters

The value to export expressed with smart parameter syntax.

Returns

Always True.

Level

Any level.

Details

Exports an evaluated smart parameter value to the Export file. If the input parameter is not a smart parameter, it will export an empty field.

Example

The example will export the value of a variable named *Expired* on field *DueDate* which is a child field on the parent page (@P) of the calling node.

```
ExportSmartParameter("@P\DueDate.Expired")
```

Parent topic: [Export actions](#)

Related reference:

[SetCSV](#)

[SetElementSeparator](#)

ExportToBatchDir

Specifies that the path for the current Export file (txt) is the Batch directory.

Syntax

```
bool ExportToBatchDir ()
```

Parameters

None.

Returns

False, if the Batch Directory is not accessible. Otherwise, True.

Level

All levels.

Details

This action sets the path for the Export file to the current Batch directory.

Attention: Usually, an Export file is placed in the application's Export folder instead of in a Batch folder of the Batch directory.

Example

```
ExportToBatchDir ()
```

Parent topic: [Export actions](#)

Related reference:

[SetExportPath](#)

Filler

Adds a string of identical filler characters to the Export.

Syntax

```
bool Filler (StrParam)
```

Parameters

Two comma-separated values. The first is a number indicating the total length in characters of the filler. The second parameter is the filler's character.

The second parameter is optional: if you do not enter a value, the action will use the most recent Global character setting.

Returns

False if the first parameter is not numeric or if the second parameter is more than 1 character. Otherwise, True.

Level

All levels.

Details

Adds a single filler character to the Export, repeated by the number of times indicated in the first input parameter. The second parameter cannot be a space. The filler string is written regardless of the data in the current field. If you wish to use a space as a filler character, use `SetSpaceFill` and then call `Filler` without the optional second parameter.

Example

```
Filler("12,n")
```

The action in the example fills the current field with 12 instances of the character "n".

Parent topic: [Export actions](#)

FixedLenLJ

Exports a specified number of characters from a field's left end (left-justified).

Syntax

```
bool FixedLenLJ (StrParam)
```

Parameters

Two comma-separated values.

1. The field's name: The name of the corresponding Field object of the Document Hierarchy.
2. The number of characters that are to be exported, counting from the field's left end.

Returns

False, if either parameter is invalid or if the action is called at the wrong level. Otherwise, True.

Level

Page level.

Details

This action is similar to the FixedLenRJ. It exports a specified number of characters from a field's left end (left-aligned).

Example

```
FixedLenLJ("Volume, 8")
```

Parent topic: [Export actions](#)

FixedLenRJ

Exports a specified number of characters from a field's right end (right-justified).

Syntax

```
bool FixedLenRJ (StrParam)
```

Parameters

Two comma-separated values.

1. The field's name: The name of the corresponding Field object of the Document Hierarchy.
2. The number of characters that are to be exported, counting from the field's right end.

Returns

False, if either parameter is invalid or if the action is called at the wrong level. Otherwise, True.

Level

Page level.

Details

Similar to the FixedLenLJ action, this action exports a specified number of characters from a field's right end (right-aligned).

Example

The following example exports 12 characters from the right end of the InvoiceDate field.

```
FixedLenRJ("InvoiceDate, 12")
```

Parent topic: [Export actions](#)

GetDATE

Exports today's Date in the format specified as the parameter.

Syntax

```
bool GetDATE (StrParam)
```

Parameters

The Date's format.

"*" stipulates the default mm/dd/yyyy construction. However, you can combine any of the following String values to define a different format:

-
- d = day of the month, 1-31
- dd = two-digit day, 01-31
- yyyy = four-digit year
- yy = two-digit year
- m = month, 1-12
- mm = two-digit month, 01-12
- ccyy = four-digit year
- y = Julian day of the year

"." and "/" are valid separators.

Returns

Always True.

Level

All levels.

Details

Exports today's Date in the format specified as the parameter.

Example

```
GetDate("*") inserts today's date into the Export file with this format:  
11/16/2005
```

Parent topic: [Export actions](#)

Related reference:

[GetTime](#)

GetProfileString

Accesses a Settings file (.ini) and adds a value in that file to your Export file.

Syntax

```
bool GetProfileString (StrParam)
```

Parameters

1. The [Section] within the Settings file.
2. The Key entry within the section, with the value you want to retrieve.
3. The name of the Settings file.

Returns

False if the settings file cannot be found. Otherwise, True. If the settings file can be found but the key entry cannot be found within the file, this action will return True.

Level

All levels.

Details

Accesses a Settings file (.ini), locates the specified key and adds the value of the key to your Export file. If the key cannot be read from the settings file, an empty string will be written to the export. If the settings file cannot be found, nothing will be written to the export.

Important: The action assumes that the Settings file resides in the current batch directory. If you want the INI file to reside in the Batches directory, specify your file name with a relative path like this: ..\myfile.ini.

Example

```
GetProfileString("General,MyValue,Batch.ini")
```

Parent topic: [Export actions](#)

GetTime

Exports the current Time in the format specified as the parameter.

Syntax

```
bool GetTime (StrParam)
```

Parameters

The parameter specifies the display format for the current time.

* = A single asterisk will use the HH:MM:SS time format. However, you can combine any of the following String values to define a different format:

-
- m = minute 1-59
- s = second 1-59
- h = 1-23
- mm, min, minute = two-digit minute, 01-59
- ss, sec, second = two-digit second, 01-59
- hh, hr, hour = two-digit hour, 01-23

"/"- are the valid separators.

Returns

Always True.

Level

Any level.

Details

Exports the current Time in the format specified by the input parameter.

Example

```
GetTime("*")
inserts the current time into the Export file with this format: 07:08:16

GetTime("hh-mm-ss")
inserts the current time into the Export file with this format: 07-08-16

GetTime("ss:mm:hh")
inserts the current time into the Export file with this format: 16:08:07
```

Parent topic: [Export actions](#)

Related reference:

[GetDATE](#)

LineItem_AddElement

Includes the specified Line Item Field object as an element of a Line Item Array.

Syntax

```
bool LineItem_AddElement (strParam)
```

Parameters

The name of the child Field object of the Document Hierarchy.

Returns

Always True.

Level

The parent field that contains the child Line Item field.

Details

This action includes the specified Line Item field object as an element of a Line Item Array.

A Line Item Array accumulates and organizes captured line item values that are retrieved from the Data file of a particular page.

Note: A rule that uses this action must be applied to the LINEITEM fields of the Document Hierarchy.

This action is used for exporting Line Item values.

Example

The following action expands the Line Item Array by one field: Price.

LineItem_ExportElements populates this element and other elements of the array with the captured values that it finds in a page's Data file before exporting them.

```
LineItem_AddElement("Price")
LineItem_ExportElements()
```

Parent topic: [Export actions](#)

Related reference:

[LineItem_BlankFields](#)

[LineItem_ClearElements](#)

[LineItem_ExportElements](#)

LineItem_BlankFields

Includes the specified number of blank fields as elements of a Line Item Array.

Syntax

```
bool LineItem_BlankFields (strParam)
```

Parameters

The number of Blank fields to export as part of the Line Item Array.

Returns

Always True.

Level

The parent field that contains the child Line Item field.

Details

This action includes the specified number of Blank Line Item fields as elements of a Line Item Array.

A Line Item Array accumulates and organizes captured line item values that are retrieved from the Data file of a particular page.

Attention: A rule that uses this action must be applied to the LINEITEM fields of the Document Hierarchy.

This action is used for exporting Blank Line Item values.

Example

The following action expands the Line Item Array by six blank fields.

LineItem_ExportElements populates this element and other elements of the array with the captured values that it finds in a page's Data file before exporting them.

```
LineItem_BlankFields("6")
```

Parent topic: [Export actions](#)

Related reference:

[LineItem_AddElement](#)

[LineItem_ClearElements](#)

[LineItem_ExportElements](#)

[LineItem_SmartParameter](#)

LineItem_ClearElements

Clears values in the Line Item Array.

Syntax

```
bool LineItem_ClearElements ()
```

Parameters

None

Returns

Always True.

Level

The parent Field object of the Document Hierarchy that contains a child Line Item field, such as you may typically find in an invoice application.

Details

This is mainly a housekeeping function you can use to be sure the array does not contain leftover values.

Example

```
LineItem_ClearElements ()
```

Parent topic: [Export actions](#)

Related reference:

[LineItem_AddElement](#)

[LineItem_BlankFields](#)

[LineItem_ExportElements](#)

[LineItem_SmartParameter](#)

LineItem_ExportElements

Exports the captured values in a page's Line Item Array that have been populated with LineItem_AddElement actions.

Syntax

```
bool LineItem_ExportElements ()
```

Parameters

None.

Returns

Always True.

Level

The parent field that contains the child Line Item sub-fields.

Details

Exports the captured values in a page's Line Item Array that have been populated with `LineItem_AddElement` actions.

Example

```
LineItem_AddElement("Price")
LineItem_AddElement("LineTotal")
LineItem_ExportElements()
NewLine()
```

This example exports the values included in the Line Item Array to your Export file.

Parent topic: [Export actions](#)

Related reference:

[LineItem_AddElement](#)

[LineItem_BlankFields](#)

[LineItem_ClearElements](#)

[LineItem_SmartParameter](#)

LineItem_SmartParameter

Add a smart parameter algorithm as an element of a Line Item Array.

Syntax

```
bool LineItem_SmartParameter (strParam)
```

Parameters

Smart Parameter to be evaluated during processing of the Line Item array.

Returns

Always True.

Level

The parent field that contains the child Line Item field.

Details

This action permits adding a smart parameter as an element of a Line Item Array to be evaluated during the Array processing.

A Line Item Array accumulates and organizes captured line item values that are retrieved from the Data file of a particular page.

Important: A rule that uses this action must be applied to the LINEITEM fields of the Document Hierarchy.

This action is used for exporting Blank Line Item values.

Example

The following action places a child field Price of the calling field node (@F) appended with the current time in format HH:MM:SS to the export file.

LineItem_ExportElements populates this element and other elements of the array with the captured values that it finds in a page's Data file before exporting them.

```
LineItem_SmartParameter("@F\Price+@TIME (HH:MM:SS) ")  
LineItem_ExportElements()
```

Parent topic: [Export actions](#)

Related reference:

[LineItem_AddElement](#)

[LineItem_BlankFields](#)

[LineItem_ClearElements](#)

[LineItem_ExportElements](#)

NewLine

Starts a new line in your Export file.

Syntax

```
bool NewLine ()
```

Parameters

None.

Returns

Always True.

Level

Any level.

Details

Starts a new line in your Export file.

Example

```
NewLine()  
Text("Export Output")
```

This sequence starts a new line and adds "Export Output" to the beginning of the line.

Parent topic: [Export actions](#)

Related reference:

[BlankLines](#)

PageVariable_ExportValue

Exports runtime values assigned to a variable of the bound Page object of the Document Hierarchy.

Syntax

```
bool PageVariable_ExportValue (StrParam)
```

Parameters

String value of the variable's name.

Returns

Always True.

Level

Page level.

Details

Exports runtime values assigned to a variable of the bound Page object of the Document Hierarchy.

Example

```
PageVariable_ExportValue ("TemplateID")
```

Parent topic: [Export actions](#)

ResetFieldVariables

Resets the variables of the bound Field object of the Document Hierarchy.

Syntax

```
bool ResetFieldVariables ()
```

Parameters

None.

Returns

Always True.

Level

Any level.

Details

This action resets some of the export settings that can be independently set by other actions.

- The default field fill character is reset to a space.
- Field alignment is reset to left.
- The ignore field status is reset to not ignore any fields that are based on the status.
- Fields are no longer set to a specific length.

Example

```
ResetFieldVariables()
```

Parent topic: [Export actions](#)

Related reference:

[SetFixedLength](#)

[SetJustified](#)

[SetZeroFill](#)

[SetFill](#)

[SetSpaceFill](#)

[SetIgnoreFieldStatus](#)

SaveFilePathAsVariable

Saves the path and name of your Export file to the variable specified by the parameter.

Syntax

```
bool SaveFilePathAsVariable (strparam)
```

Parameters

Variable name specifying where the Export file name and path will be stored.

Returns

Always True.

Level

All levels.

Details

Saves the path and name of your Export file to the variable specified by the parameter. If the variable does not exist, it will be created.

Example

```
SaveFilePathAsVariable("Export_File")
```

Parent topic: [Export actions](#)

SetCSV

Ensures that all exported values are delimited by a comma separator.

Syntax

```
bool SetCSV (StrParam)
```

Parameters

TRUE or ON: Enables CSV formatted output using a comma separator.

FALSE: Uses a custom separator, as set by `SetElementSeparator()`, between output fields.

OFF: Causes no separator to be placed between output fields.

Smart parameters are supported.

Returns

Always True.

Level

All levels.

Details

Ensures that all exported values are delimited by a comma separator.

Example

```
SetCSV("TRUE")
ExportFieldValue("Date")
ExportFieldValue("Number")
ExportFieldValue("Total")
SetCSV("FALSE")
```

This sequence will export the captured values of the Date, Number, and Total Field objects into your Export file. A comma will be added after each value to separate the fields.

Parent topic: [Export actions](#)

Related reference:
[SetElementSeparator](#)

SetElementSeparator

Ensures that all exported values are delimited by a separator designated as the parameter.

Syntax

```
bool SetElementSeparator (StrParam)
```

Parameters

The input parameter can be one of the following:

- A field separator character: Uses the provided custom character as the separator between fields. Smart parameters are supported.

- ON or TRUE: Sets the field element separator to a single space. This is the default value.
- OFF or FALSE: No separator will be placed between fields.

Smart parameters are supported.

Returns

Always True.

Level

All levels.

Details

Ensures that all exported values are delimited by a separator designated as the parameter.
Attention: If you wish to set your own custom separator, `SetCSV(FALSE)` must be called prior to exporting fields. If `SetCSV(FALSE)` is not called, then your custom element separator will not be used for export.

Example

```
SetCSV("FALSE")
SetElementSeparator("|")
```

This action uses "|" to delimit the Export file's values.

```
SetElementSeparator("Off")
```

Turns off the action.

Parent topic: [Export actions](#)

Related reference:

[SetCSV](#)

SetExportFileEncodingAsASCII

Determines if the Export file is created with ASCII or UNICODE encoding.

Member of namespace

Export

Syntax

```
bool SetExportFileEncodingAsASCII (string ASCIIEncoding)
```

Parameters

- True - Create Export files with ASCII encoding.
- False - Create Export files with UNICODE encoding (Default)

Smart parameters are supported.

Returns

Always True.

Level

Any.

Details

Determines if the Export file is created with ASCII or UNICODE encoding.

Note: This action must be called before any actions that physically write to and create the Export file. If not called, then the file is created with UNICODE encoding.

Example:

```
SetExportFileEncodingAsASCII ("True")
```

Parent topic: [Export actions](#)

SetExportPath

Specifies the path to the Export file's location. Alternatively, you can use a Smart Parameter to identify a Paths.ini file that has a set of path parameters for your application.

Syntax

```
bool SetExportPath (strParam)
```

Parameters

The complete path to the application's Export folder. Smart parameters are supported.

Returns

True, if the path specified by the parameter exists. Otherwise, False.

Level

All

Details

The action's parameter specifies the path to the Export file's location, or uses a Smart Parameter to retrieve a path's value from the application's Paths.ini file.

Example

```
SetExportPath ("c:\ParentDirectory\Invoice\Export")
```

```
SetExportPath ("@APPPATH (export) ")
```

In the second example, a smart parameter is used to obtain the location of the Export Directory from the value that is configured in the Application Manager.

Parent topic: [Export actions](#)

SetExtensionName

Assigns an extension to the current Export file.

Syntax

```
bool SetExtensionName (StrParam)
```

Parameters

The file extension you want to use, including the leading period. If you do not want a file extension, do not pass any parameter to this action. Disallowed characters are `*><[]"/\|:?` and control characters.

Smart parameters are supported.

Returns

Always True.

Level

Any level.

Details

Assigns an extension to the current Export file. If this action is not called, the default value of .TXT will be used.

Example

```
SetFileName("Export_+@BATCHID")  
SetExtensionName(".dat")
```

In this example, the Export file will have a .dat extension.

Parent topic: [Export actions](#)

Related reference:

[SetFileName](#)

SetFileName

Assigns a name to the current Export file.

Syntax

```
bool SetFileName (StrParam)
```

Parameters

The file's name (without an extension).

Smart parameters are supported.

Returns

Always True.

Level

All levels.

Details

Assigns a name to the current Export file. If `SetExtensionName` is not called, the file extension defaults to `.TXT`. If you require a different file extension, use `SetExtensionName`.

Example

```
SetFileName("Export_+@BATCHID")  
SetExtensionName(".txt")
```

This sequence establishes a series of Export files with names such as
`Export_20021231.001.txt`
`Export_20021231.002.txt`

In contrast,
`SetFileName("@BATCHID")`
`SetExtensionName(".txt")`

will establish a series of Export files with Batch IDs only:
`20021231.001.txt`
`20021231.002.txt`

Parent topic: [Export actions](#)

Related reference:

[SetExtensionName](#)

SetFill

Sets the filler character to be used to expand the current value of a field in a flat file, if the field's allowable length is greater than the length of its current export value.

Syntax

```
bool SetFill (Strparam)
```

Parameters

Single String character to be used as the filler value.

Returns

False, if more than one character is entered as a parameter. Otherwise, True.

Level

Any level.

Details

Sets the filler character to be used to expand the current value of a field in a flat file, if the field's allowable length is greater than the length of its current export value.

Attention: When using SetFill, the action SetFixedLength, FixedLenRJ or FixedLenLJ must also be used to set the maximum length of the field. You can use SetSpaceFill if you wish to make the filler character a space. Use ResetFieldVariables to clear this setting.

Example

```
SetFixedLength("10")  
SetFill("$")
```

This example sets the fill character to a \$.

Parent topic: [Export actions](#)

SetFixedLength

Uses the Numeric value you enter as a parameter to establish a fixed length of a value exported from the current field.

Syntax

```
bool SetFixedLength (StrParam)
```

Parameters

Numeric value indicating the field's length.

Returns

False if the parameter is not Numeric. Otherwise, True.

Level

Any level.

Details

Uses the Numeric value you enter as a parameter to establish a fixed length of a value exported from the current field.

Use ResetFieldVariables to clear this setting.

Example

```
SetFixedLength("12")
```

Parent topic: [Export actions](#)

Related reference:
[ResetFieldVariables](#)

SetIgnoreFieldStatus

Assigns a Numeric value to the application's SetIgnoreStatus variable. Any field with this status cannot export data to an Export file or database.

Syntax

```
bool SetIgnoreFieldStatus (Strparam)
```

Parameters

A Numeric value that represents the status of fields to be ignored by Export tasks.

Returns

False if the parameter is not Numeric. Otherwise, True.

Level

Any level.

Details

This action establishes the status that determines if an Export task will export a field's value. If the status of the field being exported matches this set value, the field will not be exported.

Use ResetFieldVariables to clear this setting.

Example

```
SetIgnoreFieldStatus ("1")
```

This example ensures that runtime values for fields with a "1" status will not be added to an Export file or update an Export database. (Typically, "1" denotes a problem field.)

Parent topic: [Export actions](#)

SetJustified

Right-justifies or left-justifies a field's exported values.

Syntax

```
bool SetJustified (StrParam)
```

Parameters

An uppercase R (right-align) or L (left-align).

Returns

False, if the parameter is not an "R" or "L". Otherwise, True.

Level

Any level.

Details

Right-justifies or left-justifies a field's exported values according to the parameter you enter.

Use `ResetFieldVariables` to clear this setting. `SetFixedLength` must also be used to set the maximum length of the field.

Example

```
SetFixedLength("10")  
SetJustified("R")
```

Parent topic: [Export actions](#)

SetOMR_Separator

For multi-punch OMR fields, uses the parameter's value as the separator character.

Syntax

```
bool SetOMR_Separator (StrParam)
```

Parameters

The separator character you want to use.

Returns

Always True.

Level

All levels.

Details

For multi-punch OMR fields, uses the parameter's value as the separator character. When multi-punch fields are exported, you typically do not want to use the same separator that you are using for fields as these values are typically all within a single field. This action allows you to specify a custom separator to be used when exporting. If this action is not called, the default value is a space.

Example

```
SetOMR_Separator(";")
```

Parent topic: [Export actions](#)

SetSpaceFill

Specifies the use of the ASCII 32 space as the global filler value to be used to expand the current value of a field in a flat file, if the field's allowable length is greater than the length of its current export value.

Syntax

```
bool SetSpaceFill ()
```

Parameters

None.

Returns

Always True.

Level

Any level.

Details

Specifies the use of the ASCII 32 space as the global filler value to be used to expand the current value of a field in a flat file, if the field's allowable length is greater than the length of its current export value.

If you use `SetSpaceFill`, the action `SetFixedLength`, `FixedLenRJ` or `FixedLenLJ` must also be used to set the maximum length of the field. Use `ResetFieldVariables` to clear this setting.

Example

```
SetSpaceFill ()
```

Note that the action specifies the use of the ASCII 32 "space" as the filler character.

Parent topic: [Export actions](#)

SetZeroFill

Sets the ASCII 48 zero as the global filler value to be used to expand the current value of a field in a flat file, if the field's allowable length is greater than the length of its current export value.

Syntax

```
bool SetZeroFill ()
```

Parameters

None.

Returns

Always True.

Level

Any level.

Details

Sets the ASCII 48 zero as the global filler value to be used to expand the current value of a field in a flat file, if the field's allowable length is greater than the length of its current export value.

SetFixedLength, FixedLenRJ or FixedLenLJ must also be used to set the maximum length of the field. Use ResetFieldVariables to clear this setting.

Example

```
SetFixedLength("10")
SetZeroFill()
```

Note that the action specifies the use of the ASCII 48 "zero" filler.

Parent topic: [Export actions](#)

Text

Places a string into the Export file.

Syntax

```
bool Text (StrParam)
```

Parameters

The string to write to the export file.

Returns

Always True.

Level

Any level.

Details

This action unconditionally adds a string to the export file. No character padding will be performed on this value.

Example

```
SetFileName("Export_+@BatchID")
SetExtensionName(".txt")
Text("This line will appear in the export file.")
```

Parent topic: [Export actions](#)

Variable_ExportValue

Exports the value assigned to a variable of the current object of the Document Hierarchy.

Syntax

```
bool Variable_ExportValue (StrParam)
```

Parameters

The name of the variable with the value you want to export.

Returns

Always True.

Level

Any level.

Details

Exports the value assigned to a variable of the current object of the Document Hierarchy.

Example

```
Variable_ExportValue("ID")
```

This action exports the value assigned to the ID property of the current object of the Document Hierarchy.

Parent topic: [Export actions](#)

Variable_IsValue

Checks to see if the parameter value matches the value assigned to a variable of the current object of the Document Hierarchy.

Syntax

```
bool Variable_IsValue (strParam)
```

Parameters

1. The name of the variable with the value you want to compare.
2. The value you want to match with the variable's value.

Returns

True if the variable's value matches the parameter's value. Otherwise, False.

Level

Any level.

Details

Checks to see if the parameter value matches the value assigned to a variable of the current object of the Document Hierarchy.

Example

```
Variable_IsValue("Invoice, Yes")
```

This action returns True if the value of the current Page object's Invoice variable is Yes.

Parent topic: [Export actions](#)

ExportDB actions

Use the ExportDB actions to set up and write information to an export database. You build the record in memory before you commit it to the database by using AddRecord.

The ExportDB actions open a connection to the database, specify the table to which the data is exported, and write the batch values to the internal data record.

- [AddRecord](#)
Inserts assembled data into the database table specified by a previous SetTableName action.
- [ExportBatchIDToColumn](#)
Exports the current Batch ID to the database column specified by the parameter.
- [ExportCloseConnection](#)
Closes an open connection to your Export database.
- [ExportFieldToColumn](#)
A page-level action that extracts the captured value of a Field object from the Data file of the current page, and specifies its target location within a table of the Export database.
- [ExportNodeXMLToColumn](#)
Exports the value of the XML property of the bound object (node) of the Document Hierarchy to a column of the Export database.
- [ExportOpenConnection](#)
Opens a connection to the database specified as the parameter.
- [ExportPropertyToColumn](#)
Adds the value of a property (variable) of the selected object to a column of the Export database
- [ExportSmartParamToColumn](#)
Adds the evaluated value of a smart parameter to a column of the Export database
- [ExportToColumn](#)
A field-level action that exports the captured value of the current Field object from the page's Data file to a target column within a previously designated table of an open Export database.
- [SetTableName](#)
Sets the name of the table in your database to which the data is to be exported.

Parent topic: [Global actions](#)

AddRecord

Inserts assembled data into the database table specified by a previous SetTableName action.

Syntax

```
bool AddRecord ()
```

Parameters

None.

Returns

False if there is no connection to the database; if an error occurs when the action attempts to add the record to the database; or if a `SetTableName` action was not previously used. Otherwise, True.

Level

All, but generally used at the Page or Field level.

Details

Inserts assembled data into the database table specified by a previous `SetTableName` action.

Important: This action must be placed after earlier actions that gather data, open the database, and access the correct table.

Example:

```
SetTableName("Invoice")
ExportFieldToColumn("VendorID, db_Vendor")
ExportFieldToColumn("Number, db_Number")
ExportFieldToColumn("Total, db_Total")
AddRecord()
```

This `ExportDB` rule applies to a Page object of the Document Hierarchy.

The actions open the database and direct the rule's attention to the Invoice table. The rule then sets up a record with three values - Vendor ID, Invoice Number and Total. Afterwards, the `AddRecord` action updates the table with new information.

Note: A return of False means the record may not have been successfully exported, and needs to be followed by a failure check rule that typically would call `SetTaskStatus(0)` (or the runner action `rr_AbortBatch`) to ensure that the batch aborts.

Parent topic: [ExportDB actions](#)

Related reference:

[SetTableName](#)

[ExportFieldToColumn](#)

ExportBatchIDToColumn

Exports the current Batch ID to the database column specified by the parameter.

Syntax

```
bool ExportBatchIDToColumn (strParam)
```

Parameters

Three comma-separated values:

1. The name of the database column that will hold the value.
2. The name of the field whose value will be stored in the column.
3. Optional: Can be set to the numerical value of 1, 2, or 3.

If this parameter is not specified, quotes will be placed around the value in the SQL query to update the table.

Setting the value to 1 causes this action to return False if the Table name is invalid; otherwise the error will not be discovered until the action that performs the DB insert call, such as an AddRecord action, is called. This value would typically only be used in a development environment as it will increase processing time.

Setting this value to 2 allows NULL column values and inserts no quotes around values (for numeric) in the SQL query to update the database.

Setting the value to 3 allows NULL and if the column value is not NULL, the column value will be surrounded in quotes in the SQL query to update the database.

Returns

False if the second parameter is set to "1" and the parameters do not identify a valid database column. Otherwise, True.

Level

All, but generally at the Page or Field level.

Details

Exports the current Batch ID to the database column specified by the parameter. The action also has an optional Style parameter with two values.

Example:

```
SetTableName("Export_Results")
ExportBatchIDToColumn("db_BatchID,1,2")
ExportFieldToColumn("Date,db_Date")
ExportFieldToColumn("Total,db_Total")
AddRecord()
```

Parent topic: [ExportDB actions](#)

Related reference:

[AddRecord](#)

[ExportFieldToColumn](#)

[ExportPropertyToColumn](#)

ExportCloseConnection

Closes an open connection to your Export database.

Syntax

```
bool ExportCloseConnection ()
```

Parameters

None.

Returns

True, even if the connection is already closed.

Level

All, but generally used as part of a separate RuleSet at the Batch level.

Details

Closes an open connection to the previously opened Export database.

Usually, this action is placed in a RuleSet that is separate from the RuleSet that opens the connection and stores the data.

For example this action could be placed into a RuleSet called ExportDBCclose, and attached to a batch level close event which executes after all data has been exported from the batch to the specified database.

Example:

```
ExportCloseConnection()
```

This action closes the previously opened connection to the Export database.

This action is usually part of a separate RuleSet that prevents the need to repeatedly open the connection to the Export database. (You can open the connection once in the first RuleSet, export data from all documents and pages in the batch, then close the connection once in the second RuleSet.)

Parent topic: [ExportDB actions](#)

Related reference:

[ExportOpenConnection](#)

ExportFieldToColumn

A page-level action that extracts the captured value of a Field object from the Data file of the current page, and specifies its target location within a table of the Export database.

Syntax

```
bool ExportFieldToColumn (strParam)
```

Parameters

Three comma-separated values:

1. The name of the field whose value will be stored in the column.
2. The name of the database column that will hold the value.
3. Optional: Can be set to the numerical value of 1, 2, or 3.

If this parameter is not specified, quotes will be placed around the value in the SQL query to update the table.

Setting the value to 1 causes this action to return False if the Table name is invalid; otherwise the error will not be discovered until the action that performs the DB insert call, such as an AddRecord action, is called. This value would typically only be used in a development environment as it will increase processing time.

Setting this value to 2 allows NULL column values and inserts no quotes around values (for numeric) in the SQL query to update the database.

Setting the value to 3 allows NULL and if the column value is not NULL, the column value will be surrounded in quotes in the SQL query to update the database.

Returns

False if:

1. There is no connection to the database.
2. The database column specified as a parameter does not exist.
3. The Field object identified by the parameter does not exist.
4. A SetTableName action was not used previously.

Otherwise, True.

Level

Page level only.

Details

This is a page-level action that extracts the captured value of a Field object from the Data file of the current page, and specifies its target location within a table of the Export database.

Within a rule, this action should run before an AddRecord action, which commits the data to the database.

Example:

```
SetTableName("Export_Results")
ExportFieldToColumn("VendorID,db_Number,2")
ExportFieldToColumn("Date,db_Date")
ExportFieldToColumn("Total,db_Total")
AddRecord()
```

This action exports the captured value of three Field objects from the Data file of the current page, to corresponding columns of the Export_Results table, within the open Export database. It also provides the action with an optional Style value.

Important: Make sure you use the ExportOpenConnection action to establish a connection to your Export database. This is usually accomplished by a rule at the Batch level.

Parent topic: [ExportDB actions](#)

Related reference:

[SetTableName](#)

[ExportToColumn](#)

ExportNodeXMLToColumn

Exports the value of the XML property of the bound object (node) of the Document Hierarchy to a column of the Export database.

Syntax

```
bool ExportNodeXMLToColumn (strParam)
```

Parameters

Three comma-separated values:

1. The smart parameter path to the bound object of the Document Hierarchy. This object's XML property will be added to a designated column of the Export database.
2. The String value of the name of the target column in the Export database. The action will add the value of the calling object's XML property to this column.
3. Optional: Can be set to the numerical value of 1 or 3. If this parameter is not specified, quotes will be placed around the value in the SQL query to update the table.

Setting the value to 1 causes this action to return False if the Table name is invalid; otherwise the error will not be discovered until the action that performs the DB insert call, such as an AddRecord action, is called. This value would typically only be used in a development environment as it will increase processing time.

Setting the value to 3 allows NULL and if the column value is not NULL, the column value will be surrounded in quotes in the SQL query to update the database.

Returns

False if:

1. There is no connection to the database.
2. The column identified by the parameter does not exist.
3. A SetTableName action was not previously used.
4. The smart parameter path does not point to a valid object of the Document Hierarchy.

Otherwise, True.

Level

All, but generally at the Page or Field level.

Details

This action exports the value of the XML property of the bound object (node) of the Document Hierarchy to a column of the Export database.

Example:

```
ExportNodeXMLToColumn("@P\MyField,MYDBCOLUM")
```

Parent topic: [ExportDB actions](#)

ExportOpenConnection

Opens a connection to the database specified as the parameter.

Syntax

```
bool ExportOpenConnection (strParam)
```

Parameters

The Connection String of the target export database. Use Smart parameters to avoid clear text passwords in your application.

Returns

True, if the connection opens. Otherwise, False.

Level

All, but most often used at the Batch level.

Details

Opens a connection to the database specified as the parameter.

A rule that contains this action can apply to any object of the Document Hierarchy, but it is most often used at the batch level.

Example:

```
ExportOpenConnection ("@APPVAR (values/dsn/exportdb:cs) ")
```

Note: This action must come before any other ExportDB actions. This example uses smart parameters to obtain the connection string from the application service where the custom connection string key `exportdb` was created in the Custom Values tab. It provides a single location for the connection string and security for passwords.

Note: If the action is establishing a connection with an Oracle database, or a SQL Server database using SQL Server Authentication, be sure to expand the DSN parameter by adding the correct Provider, user ID and Password. For example:

- Oracle:

```
ExportOpenConnection ("PROVIDER=ODBCORACLE;DSN=1040Look;CATALOG=;DBNTA=;UID=Admin;PWD=Admin;")
```
- SQL Server Authentication:

```
ExportOpenConnection ("PROVIDER=ODBCMSSQL;DSN=1040Look;CATALOG=;DBNTA=;UID=Admin;PWD=Admin;")
```

Note: Although the Oracle and SQL Server examples show the connection string, it is recommended that a smart parameter is used to obtain the connection string from the application service, as shown in the first example, to provide password security and application portability.

Parent topic: [ExportDB actions](#)

Related reference:

[ExportCloseConnection](#)

ExportPropertyToColumn

Adds the value of a property (variable) of the selected object to a column of the Export database

Syntax

```
bool ExportPropertyToColumn (strParam)
```

Parameters

Three comma-separated values:

1. The property name whose value will be stored in the column.
2. The name of the database column that will hold the value.
3. Optional: Can be set to the numerical value of 1, 2, or 3. If this parameter is not specified, quotes will be placed around the value in the SQL query to update the table.

Setting the value to 1 causes this action to return False if the Table name is invalid; otherwise the error will not be discovered until the action that performs the DB insert call, such as an AddRecord action, is called. This value would typically only be used in a development environment as it will increase processing time.

Setting this value to 2 allows NULL column values and inserts no quotes around values (for numeric) in the SQL query to update the database.

Setting the value to 3 allows NULL and if the column value is not NULL, the column value will be surrounded in quotes in the SQL query to update the database.

Returns

False if:

1. There is no connection to the Export database.
2. The column of the database does not exist.
3. The property (variable) identified by the parameter does not exist.
4. A SetTableName action was not used previously.

Otherwise, True.

Level

All, but generally used at the Page or Field level.

Details

Adds the value of a property (variable) of the selected object to a column of the Export database.

Example:

```
SetTableName ("Export_Results")
ExportFieldToColumn ("VendorID, db_Number")
ExportPropertyToColumn ("Status, db_PageStatus, 2")
ExportFieldToColumn ("Total, db_Total")
AddRecord ()
```

This sequence updates the db_PageStatus column of the Export_Results table with the value of the selected object's Status property. It also uses the optional third parameter to define a Style for the action.

Parent topic: [ExportDB actions](#)

Related reference:

[ExportToColumn](#)

[ExportFieldToColumn](#)

[ExportBatchIDToColumn](#)

ExportSmartParamToColumn

Adds the evaluated value of a smart parameter to a column of the Export database

Syntax

```
bool ExportSmartParamToColumn (strParam)
```

Parameters

Three comma-separated values:

1. A smart parameter that specifies the value to be stored into the column.
2. The name of the database column that will hold the value.
3. Optional: Can be set to the numerical value of 1, 2, or 3. If this parameter is not specified, quotes will be placed around the value in the SQL query to update the table.

Setting the value to 1 causes this action to return False if the Table name is invalid; otherwise the error will not be discovered until the action that performs the DB insert call, such as an AddRecord action, is called. This value would typically only be used in a development environment as it will increase processing time.

Setting this value to 2 allows NULL column values and inserts no quotes around values (for numeric) in the SQL query to update the database.

Setting the value to 3 allows NULL and if the column value is not NULL, the column value will be surrounded in quotes in the SQL query to update the database.

Returns

False if:

1. There is no connection to the database.
2. The column identified by the parameter does not exist.
3. A SetTableName action was not previously used.

Otherwise, True.

Level

All, but generally at the Page or Field level.

Details

Using the database opened by an earlier ExportDB action, this action will store a value into the specified column for the current row.

The action allows the input value to be specified with a smart parameter.

Example:

```
ExportSmartParamToColumn("@P\MyField.TYPE,EXPDBCOLUMN")
```

Parent topic: [ExportDB actions](#)

ExportToColumn

A field-level action that exports the captured value of the current Field object from the page's Data file to a target column within a previously designated table of an open Export database.

Syntax

```
bool ExportToColumn (strParam)
```

Parameters

Two comma-separated values:

1. The name of the database column that will hold the value of the current field.
2. Optional: Can be set to the numerical value of 1, 2, or 3. If this parameter is not specified, quotes will be placed around the value in the SQL query to update the table.

Setting the value to 1 causes this action to return False if the Table name is invalid; otherwise the error will not be discovered until the action that performs the DB insert call, such as an AddRecord action, is called. This value would typically only be used in a development environment as it will increase processing time.

Setting this value to 2 allows NULL column values and inserts no quotes around values (for numeric) in the SQL query to update the database.

Setting the value to 3 allows NULL and if the column value is not NULL, the column value will be surrounded in quotes in the SQL query to update the database.

Returns

False if:

1. There is no connection to the database.
2. The column identified by the parameters does not exist.
3. A SetTableName action was not previously used.

Otherwise, True.

Level

Field level only.

Details

This field-level action exports the captured value of the current Field object from the page's Data file to a target column within a previously designated table of an open Export database. Optionally, it can assign a Style to the action.

Example:

```
Batch Level:  
ExportOpenConnection ("@APPVAR(values/dsn/exportdb:cs")  
SetTableName ("Export_Results")
```

```
Current field:  
ExportToColumn ("db_Date, 3")
```

```
Last Field:  
AddRecord ()
```

This example exports the captured value of the Field object to which the rule applies, from the Data file of the current page to the db_Date column of the Export_Results table, within the Export database.

It also uses the second, optional parameter value to assign a Style to the action.

Parent topic: [ExportDB actions](#)

Related reference:

[ExportFieldToColumn](#)

[ExportPropertyToColumn](#)

SetTableName

Sets the name of the table in your database to which the data is to be exported.

Syntax

```
bool SetTableName (strParam)
```

Parameters

Two comma separated parameters:

1. The name of the database table where the exported data will be inserted.
2. Optional: If the numeric value of 1 is specified, the action will immediately check the database to confirm that the supplied table name is valid.

If the table name is not valid, SetTableName will return False. Note that this value causes an extra database call so it is typically specified only during development, not in a production system as it will increase the number of database calls and running time.

If the second parameter is not specified, this action will always return True.

If the specified table name is in valid, an error will be returned when the current row is inserted to the database on the next AddRecord call.

Returns

Returns False only if the first parameter contains an invalid table name and a value of 1 is specified as a second parameter. Otherwise this action always returns True.

Level

All.

Details

Sets the name of the table in your database to which the data is to be exported. This action needs to be used before the AddRecord action.

Example:

```
SetTableName(Export_Results,1)
ExportFieldToColumn(MyDate, db_Date)
AddRecord()
```

Parent topic: [ExportDB actions](#)

Related reference:

[AddRecord](#)

ExportXML actions

Use the ExportXML actions to set up and write information to an export XML file.

The ExportXML actions can create nodes, assign attribute values to nodes, and specify the path to the storage location and the name of the export XML file.

- [xml_CommitNode](#)
Commits (closes) current xml node with the xml tag value of the parameter.
- [xml_NewNode](#)
Creates a child node under the specified parent node, creating the parent node if necessary.
- [xml_SaveFile](#)
Commits all unsaved nodes and saves the XML file to disk. Nodes that were not committed using the [xml_CommitNode](#) action will be committed to the parent node when [xml_SaveFile](#) is called, and will appear subsequent to nodes previously committed using the [xml_CommitNode](#) action.
- [xml_SetAttributeValue](#)
Assigns attributes to a specific node.
- [xml_SetExportPath](#)
Specifies the full path to the directory that will store the export XML file.
- [xml_SetFileEncodingAsASCII](#)
Determines if the Export file is created with ASCII or UNICODE encoding.
- [xml_SetFileName](#)
Specifies the name for the export XML file, which does not include the .xml extension.
- [xml_SetNodeValue](#)
Sets the value of the specified node.

Parent topic: [Global actions](#)

xml_CommitNode

Commits (closes) current xml node with the xml tag value of the parameter.

Syntax

```
bool xml_CommitNode (StrParam)
```

Parameters

String value of the xml Tag. Smart parameters are supported.

Returns

Always True.

Level

All.

Details

This action closes the specified node, which allows a new node with the same tag to be created at the same hierarchical level in the output xml. This action can use Smart Parameters.

Remember: Use of this action can change the expected order of xml nodes as they appear in the final xml, since this action commits the current xml to the parent xml node when it is called. Sibling xml nodes that were not committed using this action are later committed to the parent node when the `xml_SaveFile` action is called, so those xml nodes will appear after nodes previously committed using the `xml_CommitNode` action.

Example

```
xml_CommitNode("LineTotal")
```

Example commits (closes) the current xml node with a `Tag Line Total`.

Parent topic: [ExportXML actions](#)

xml_NewNode

Creates a child node under the specified parent node, creating the parent node if necessary.

Syntax

```
bool xml_NewNode (Strparam)
```

Parameters

Comma-separated String values of:

1. The NodeID (tag name) of a new child node.
2. The NodeID (tag name) of the parent node, if there is a parent.

Smart parameters are supported.

Returns

True, if parent node exists. False, if duplicate root node is declared, or parent NodeID does not exist.

Level

All.

Details

The new NodeID followed by the parent NodeID creates a new Node in the Export XML file. The action can use Smart Parameters.

The first `xml_NewNode` action in a rule set creates the root node of the XML file using the new child NodeID. The parent NodeID must be blank.

Example

```
xml_SetExportPath("C:\ParentDir\APT")
ml_SetFileName("@BATCHID")
xml_NewNode("B")
ml_SetAttributeValue("B, id, @BATCHID")
```

The example starts an export XML file that is named the same as the current batch ID. It has a root node `B` whose `id` is assigned the batch ID. It is expected that more rules continue to build the XML structure. The last action that was called must call `xml_SaveFile` to save the export file to disk.

```
xml_NewNode ("ClaimsData, HCFA")
```

Duplicated NodeIDs cause the previous node of the same ID to commit to its specific parent node and is no longer available for modification. Adding a second child NodeID with the same Tag name of the root node causes the action to return False.

Parent topic: [ExportXML actions](#)

Related reference:

[xml_SaveFile](#)

xml_SaveFile

Commits all unsaved nodes and saves the XML file to disk. Nodes that were not committed using the xml_CommitNode action will be committed to the parent node when xml_SaveFile is called, and will appear subsequent to nodes previously committed using the xml_CommitNode action.

Syntax

```
bool xml_SaveFile ()
```

Parameters

None.

Returns

True if the file is created successfully. Otherwise, False.

Level

Batch, Document or Page level.

Details

You must use this action to complete the creation of the export XML file.

Example

```
xml_SaveFile ()
```

Parent topic: [ExportXML actions](#)

xml_SetAttributeValue

Assigns attributes to a specific node.

Syntax

```
bool xml_SetAttributeValue (StrParam)
```

Parameters

Three comma-separated values:

1. The NodeID.
2. The attribute's name.
3. The value to be assigned to the attribute.

Smart parameters are supported.

Returns

False, if the node does not exist. Otherwise, True.

Level

All.

Details

Sets an attribute value within a specific node in the XML hierarchy.

Example

The following example assigns the current Page's Number field value to the Xpage node's Number attribute.

```
xml_SetAttributeValue ("Xpage, Number, @P\Number")
```

The following example shows how creation of an XML for invoice line items might look.

```
xml_NewNode ("LineItem, Invoice")
xml_SetAttributeValue ("LineItem, id, @ID")
xml_NewNode ("ItemID, LineItem")
xml_SetNodeValue ("ItemID, @F\ItemID")
xml_NewNode ("ItemDesc, LineItem")
xml_SetNodeValue ("ItemDesc, @F\ItemDesc")
xml_NewNode ("Qty, LineItem")
xml_SetNodeValue ("Qty, @F\Qty")
xml_CommitNode ("LineItem")
```

Parent topic: [ExportXML actions](#)

xml_SetExportPath

Specifies the full path to the directory that will store the export XML file.

Syntax

```
bool xml_SetExportPath (strParam)
```

Parameters

Specifies the full path to the directory that stores the export XML file. If this action is not called, the XML file is saved in the batch directory. Smart parameters are supported.

Returns

Always **True**.

Level

All.

Details

This action will only set the path. You will still need to use `xml_SetFileName` to set the name of the export file.

Example

```
xml_SetExportPath("C:\Invoice\ExportXML")
xml_SetFileName("@BatchID")
```

Parent topic: [ExportXML actions](#)

xml_SetFileEncodingAsASCII

Determines if the Export file is created with ASCII or UNICODE encoding.

Member of namespace

ExportXML

Syntax

```
bool xml_SetFileEncodingAsASCII ( string ASCIIEncoding)
```

Parameters

- True - Create Export files with ASCII encoding.
- False - Create Export files with UNICODE encoding (Default)

Smart parameters are supported.

Returns

Always True.

Level

Any.

Details

Determines if the Export file is created with ASCII or UNICODE encoding.

Note: This action must be called before any actions that physically write to and create the Export file. If not called, then the file is created with UNICODE encoding.

Example:

```
xml_SetFileEncodingAsASCII("True")
```

Parent topic: [ExportXML actions](#)

xml_SetFileName

Specifies the name for the export XML file, which does not include the .xml extension.

Syntax

```
bool xml_SetFileName (StrParam)
```

Parameters

The name of the XML export. Smart Parameters are supported.

Returns

Always True.

Level

All.

Details

This action is required to set the name of the XML Export file. This action can be called at any level, but it is usually called at the batch level. If the `xml_SetExportPath` action is not called, the file is saved in the batch folder.

Example

```
xml_SetExportPath("C:\Invoice\ExportXML")
xml_SetFileName("@BatchID")
```

Parent topic: [ExportXML actions](#)

Related reference:

[xml_SaveFile](#)

xml_SetNodeValue

Sets the value of the specified node.

Syntax

```
bool xml_SetNodeValue (StrParam)
```

Parameters

Two comma-separated values:

1. The NodeID.
2. The value to assign to the node.
3. The action defaults the current object's value.

Smart parameters are supported.

Returns

Always True.

Level

All.

Details

Assigns a value to a specific node. The action can use Smart Parameters.

Example

The following example shows how creation of an XML for invoice line items might look.

```
xml_NewNode("LineItem, Invoice")
xml_SetAttributeValue("LineItem, id, @ID")
xml_NewNode("ItemID, LineItem")
xml_SetNodeValue("ItemID, @F\ItemID")
xml_NewNode("ItemDesc, LineItem")
xml_SetNodeValue("ItemDesc, @F\ItemDesc")
xml_NewNode("Qty, LineItem")
xml_SetNodeValue("Qty, @F\Qty")
xml_CommitNode("LineItem")
```

Parent topic: [ExportXML actions](#)

Related reference:

[xml_NewNode](#)

FileIO actions

Use the FileIO actions to do various file system functions.

The FileIO actions can check for available disk space, copy or delete directories, and write file values to a specified variable.

- [CheckFreeDiskSpace](#)
Checks the size of the available disk space.
- [CopyDirectory](#)
Copies a directory and its subdirectories
- [CopyFile](#)
Copies a file.
- [DeleteDirectory](#)
Deletes a directory and optionally deletes subdirectories
- [DeleteFile](#)
Deletes a file.
- [GetFileSize](#)
Obtains the size of a file and stores it in the specified variable.
- [GetProfileString](#)
Reads a key value from a settings file.
- [IsDirectoryPresent](#)
Determines if the specified directory exists and optionally creates it.
- [IsFilePresent](#)
Determines if the specified file exists.

- [IsFileReadOnly](#)
Tests the read only attribute of a file.
- [IsProfilePresent](#)
Tests that a profile exists and that a specific section and key exists within it.
- [Readtextfile](#)
Reads a text file.
- [RenameFile](#)
Renames or moves the specified file.
- [SetFileReadOnly](#)
Sets or removes the read only attribute from a file or set of files.
- [SetProfileString](#)
Writes a value to a profile file, typically called an INI file.
- [SplitFileName](#)
Splits a file name into user specified variables.
- [ZipOcrResults](#)
Creates a .zip file to store recognition results for every page of the document.

Parent topic: [Global actions](#)

CheckFreeDiskSpace

Checks the size of the available disk space.

Member of namespace

FileIO

Syntax

```
bool CheckFreeDiskSpace (string DriveLetter, string Threshold, string  
TargetVariable)
```

Parameters

string DriveLetter

string Threshold

string TargetVariable

Parameters

DriveLetter: The drive letter to test for available disk space. UNC paths are not supported.

Threshold: Optional. The minimum amount of required disk space in bytes. If provided, this value must be a positive integer.

TargetVariable: Optional. The DCO variable to store the value of the available disk space.

Smart parameters are supported for each parameter.

Returns

False If the threshold is specified and the amount of free disk space is less than the specified value. If the target variable is provided but cannot be set or if the drive letter is invalid. Otherwise, True.

Level

Any level.

Details

Checks the available amount of free disk space in bytes.

Example

```
CheckFreeDiskSpace ("D", "1000", "@P.FreeSpace")
```

Checks that the amount of available disk space on drive D is at least 1000 bytes. It also stores the amount of free space in the page level runtime DCO variable *FreeSpace*.

```
CheckFreeDiskSpace ("D", "@B.MyLimit", "")
```

Only checks that the amount of available disk space on drive D is at least as much as the value specified in the batch level variable *MyLimit*.

```
CheckFreeDiskSpace ("D", "", "@P.FreeSpace")
```

Stores only the amount of free space in the page level runtime DCO variable *FreeSpace*.

Parent topic: [FileIO actions](#)

CopyDirectory

Copies a directory and its subdirectories

Member of namespace

FileIO

Syntax

```
bool CopyDirectory (string SourceDirectory, string DestDirectory, bool Recursive)
```

Parameters

SourceDirectory
Type: string
DestDirectory
Type: string
Recursive
Type: bool

Parameters

- SourceDirectory : The path of the directory to copy.
- DestDirectory : The path to the destination directory.

- Recursive : True copies the directory files and subdirectories. False only copies the files in the initial directory.

Smart parameters are supported for the directory paths.

Returns

False if a failure occurs when copying a directory. If some of the files have been copied prior to the failure, those files remain. Otherwise, True.

Level

Any level.

Details

Copies the source directory to the target directory. If the target directory does not exist, it will be created, however the parent directory does need to exist.

Example

```
CopyDirectory("C:\MyDir1\MyDir2", "C:\MyNewDir", True)
```

This example copies MyDir2 and all files and subdirectories to C:\MyNewDir.

Parent topic: [FileIO actions](#)

CopyFile

Copies a file.

Member of namespace

FileIO

Syntax

```
bool CopyFile (string sourcefile, string targetfile, bool overwrite)
```

Parameters

string sourcefile

string targetfile

bool overwrite

Parameters

sourcefile: The name and path of the file to copy. Smart parameters are supported.

targetfile: The target path and file name. Smart parameters are supported.

overwrite: If the target file exists, this parameter determines whether it is overwritten. True overwrites the target file.

Returns

True if the file is successfully copied. Otherwise, False.

Level

All levels.

Details

The file is copied from one location to the specified location. If a file of the same target name exists in the target directory, the overwrite parameter determines whether that file is overwritten by the new file. The output directory must exist for this action to succeed. You can use action `IsDirectoryPresent` to create the target directory.

If target file ends with a backslash, meaning only the target directory is specified, the same file name from the source string is used as the target file.

Because the target file name can be different from the source file, this action allows the copied file to be renamed in one step. If you want to do a "move" operation, call the `DeleteFile` action to remove the source file.

Smart parameters are supported in the source and target file name parameters. DOS * and ? wildcards are not supported.

Example

```
CopyFile("C:\MyFile.txt", "c:\temp\+@BATCHID+.txt", true)
```

This action copies `MyFile.txt` to the temp directory and give it a new name of the current batch ID.

```
CopyFile("@VAR(IMAGEFILE)", "c:\temp\", true)
```

This action copies the file that is specified by the variable `IMAGEFILE` and copies it to the temp directory.

Parent topic: [FileIO actions](#)

DeleteDirectory

Deletes a directory and optionally deletes subdirectories

Member of namespace

FileIO

Syntax

```
bool DeleteDirectory (string Directory, bool Recursive, bool FailureReturnValue)
```

Parameters

string Directory

bool Recursive

bool FailureReturnValue

Parameters

Directory: The path of the directory to delete.

Recursive: True deletes the directory and all files and subdirectories. False deletes only the directory and requires that the directory is empty.

FailureReturnValue: True causes the action to always return True, even if the delete fails. False causes the action to return False if the directory delete fails.

Smart parameters are supported for the directory path.

Returns

Always returns True if the directory is deleted.

If a failure occurs when you delete a directory, the action returns the value of FailureReturnValue. This result lets you determine whether the action returns True or False on a directory failure. Depending on the application, it might make sense to ignore a delete failure so this action can be set to always return True.

Level

Any level.

Details

Deletes the specified directory and can optionally delete subdirectories.

Example

```
DeleteDirectory("C:\MyDir1\MyDir2", True, True)
```

This example deletes MyDir2 and all files and subdirectories. If the delete fails, the action still returns True.

Parent topic: [FileIO actions](#)

DeleteFile

Deletes a file.

Member of namespace

FileIO

Syntax

```
bool DeleteFile (string filename)
```

Parameters

filename

Type: string

Parameters:

filename: The name and path of the file to delete. Smart parameters are supported.

Returns

Always True.

Level

All levels.

Details

Deletes the specified file. DOS wildcards are permitted in the file name.

Example

The following example deletes the specific file.

```
DeleteFile("C:\Temp\DeleteThis.txt")
```

The following example deletes all of the files in the Temp directory.

```
DeleteFile("C:\Temp\*. *")
```

The following example deletes the file name that matches the current batch ID and has an extension of TXT.

```
DeleteFile("C:\Temp\@BATCHID+.txt")
```

Parent topic: [FileIO actions](#)

GetFileSize

Obtains the size of a file and stores it in the specified variable.

Member of namespace

FileIO

Syntax

```
bool GetFileSize (string filename, string targetVariable)
```

Parameters

filename

Type: string

targetVariable

Type: string

Parameters

- filename : The name and path of the file.
- targetVariable : The name of the variable to store the file size. The variable level must be specified with a Smart Parameter.

Smart parameters are supported for each of the input parameters.

Returns

False if the target variable is blank. Otherwise, True.

Level

All levels.

Details

The size of the specified file will be stored into the specified variable. This can be helpful to test for problems, such as empty files. Subsequent actions can perform tests on the stored value and act upon it.

If the specified file is not found, the action will return True and a size of 0 will be stored into the variable. If any other kind of error occurs, the action will return False and the variable may or may not be set to 0.

Example

```
GetFileSize("C:\Temp\MyFile.txt", "@B.MyVariable")
```

This example sets the file size in the variable *MyVariable* at the batch level.

Parent topic: [FileIO actions](#)

GetProfileString

Reads a key value from a settings file.

Member of namespace

FileIO

Syntax

```
bool GetProfileString (string filename, string section, string key, string targetVariable)
```

Parameters

filename
Type: string

section
Type: string

key
Type: string

targetVariable
Type: string

Parameters

- filename : The INI file name.
- section : The section within the file.
- key : The key within the section.
- targetVariable : The variable where to store the value. The variable level must be specified with a smart parameter.

Smart parameters are supported for each of the input parameters.

Returns

True if the settings file exists and the target variable is valid. Otherwise, False.

Level

All levels.

Details

Reads the value from a settings file, typically a .INI file, and stores it into the variable specified. If the variable does not exist, it will be created. If the key does not exist but the file does exist, the action will still return true and an empty string will be stored in the variable. If you need to test for a blank key value, you can use the `IsProfilePresent` action.

If you are reading Unicode characters from an INI file, it is required that the file is in a format that supports Unicode. If the file is not a Unicode format, the Unicode characters may appear incorrectly when read by `GetProfileString` or displayed in an editor. Most file editors will let you choose the type of file encoding. See the help for your specific editor.

Example

```
GetProfileString("C:\Settings.ini", "MySection", "MyKey", "@B.MyVariable")
```

The value for MyKey is placed into the variable *MyVariable*.

Parent topic: [FileIO actions](#)

IsDirectoryPresent

Determines if the specified directory exists and optionally creates it.

Member of namespace

FileIO

Syntax

```
bool IsDirectoryPresent (string directoryName, bool create, bool testExistence)
```

Parameters

directoryName
Type: string

create
Type: bool
testExistence
Type: bool

Parameters

- `directoryName` : The directory path to test. Smart parameters are allowed.
- `create` : Specifies if the directory should be created. True creates the directory, if it does not exist.
- `testExistence` : Determines if True should be returned if the directory exists or does not exist.

Returns

If `testExistence` is True, the action will return True if the directory exists or if the directory did not exist but was successfully created.

If `testExistence` is False, the action will return True if the directory does not exist. This allows you to perform negative tests that will return true when a directory does not exist.

If an error occurs, the action will return False.

Level

All levels.

Details

Checks for the existence of a directory. Depending on the input parameters, if the directory does not exist, the action creates the directory. The meaning of the return value can be changed using the `testExistence` parameter. This allows rules to continue if a directory exists or rules can continue if a directory does not exist.

Example

```
IsDirectoryPresent("c:\temp", true, true)
```

This example creates the directory `C:\temp`, if it does not exist, and returns True if the directory exists or was created successfully.

```
IsDirectoryPresent("c:\temp", true, false)
```

This example creates the directory `C:\temp`, if it does not exist, and returns False if the directory exists or was created successfully.

Parent topic: [FileIO actions](#)

IsFilePresent

Determines if the specified file exists.

Member of namespace

FileIO

Syntax

```
bool IsFilePresent (string filename, bool testExistence)
```

Parameters

filename
Type: string
testExistence
Type: bool

Parameters

- filename : The file name and path. Smart parameters are allowed.
- testExistence : true tests that the file exists, false tests that the file does not exist.

Returns

If testExistence is True, the action will return True if the file exists. If testExistence is False the action will return True if the file does not exist. If an error occurs, the action will return False.

Level

All levels.

Details

Checks for the existence of a file. Depending on the testExistence parameter, the action can return true if a file exists or true if a file does not exist to provide flexibility when creating rules.

Example

```
IsFilePresent("C:\MyDir\MyFile.abc", false)
```

In this example, if the file does not exist, the action returns True and any subsequent actions are processed.

Parent topic: [FileIO actions](#)

IsFileReadOnly

Tests the read only attribute of a file.

Member of namespace

FileIO

Syntax

```
bool IsFileReadOnly (string filename, bool testForReadOnly)
```

Parameters

filename
Type: string

testForReadOnly
Type: bool

Parameters

- filename : The file name and path of the file to test. Smart parameters are allowed.
- testForReadOnly : Determines if the action should return true for read-only or read-write.

Returns

If testForReadOnly is True, the action returns True if the file is read only.

If testForReadOnly is False, the action returns True if the file is not read only.

If an error occurs, or if the file does not exist, the action returns False.

Level

All levels.

Details

This action will indicate if a file is set to read only.

Example

```
IsFileReadOnly("c:\mydir\myfile.txt", true)
```

if the file myfile.txt is set to read only, the action returns True.

```
IsFileReadOnly("c:\mydir\myfile.txt", false)
```

if the file myfile.txt is not set to read only, the action returns True.

Parent topic: [FileIO actions](#)

IsProfilePresent

Tests that a profile exists and that a specific section and key exists within it.

Member of namespace

FileIO

Syntax

```
bool IsProfilePresent (string filename, string section, string key, bool  
testExistence)
```

Parameters

filename
Type: string
section
Type: string

key
Type: string
testExistence
Type: bool

Parameters

- filename : The INI file name.
- section : The section within the file.
- key : The key within the section.
- testExistence : true tests that the it exists, false tests that it does not exist.

The filename, section, and key parameters can accept smart parameters.

Returns

True if testExistence is true and the section and key is found within the profile and the key has an assigned value. True if testExistence is false and the section or key is not found within the profile or the key value is blank. Otherwise, False.

Level

All levels.

Details

Checks that a specific section and key exists within a settings file, typically an INI file. It does not test the value of the key, only that a value exists.

Example

```
IsProfilePresent("C:\MyDir\settings.ini", "mysection", "mykey", true)
```

In this example, if settings.ini exists and contains *mysection*, *mykey* and if *mykey* has a non-blank value, it returns True.

Parent topic: [FileIO actions](#)

Readtextfile

Reads a text file.

Member of namespace

FileIO

Syntax

```
bool ReadTextFile (string filename, string target)
```

Parameters

- filename - The name and path of the text file to read.

- target - The DCO node to store the value of the text file.

Smart parameters are supported for each parameter.

Returns

True if the text file exists and was successfully read, otherwise False.

Level

All levels.

Details

Reads the specified text file.

Example:

```
ReadTextFile("C:\Temp\ReadThis.txt", "@P.text")
```

Parent topic: [FileIO actions](#)

RenameFile

Renames or moves the specified file.

Member of namespace

FileIO

Syntax

```
bool RenameFile (string oldName, string newName, bool overwrite)
```

Parameters

oldName
 Type: string
newName
 Type: string
overwrite
 Type: bool

Parameters

- oldName : The source file name and path. Allows smart parameters.
- newName : The destination file name and path. Allows smart parameters.
- overwrite : Specifies if the destination file should be overwritten. True overwrites an existing file.

Returns

True if the file is successfully renamed or moved. Otherwise, False.

Level

All levels.

Details

Renames the specified file to the new name. If the directories for the original and new file names are different, the file will be moved to the new directory. If the overwrite parameter is true, the file will overwrite an existing file. If overwrite is false, an existing file will not be overwritten.

Example

Code	Comment
<pre>RenameFile("C:\MyDir\File1.txt", "C:\MyDir\File2.txt", true)</pre>	Renames the file within the same directory. The renaming overwrites any existing file File2.txt.
<pre>RenameFile("C:\MyDir\File1.txt", "C:\New\File1.txt", true)</pre>	Moves the file from the MyDir directory to the New directory. The move overwrites any existing file File1.txt.
<pre>RenameFile("@DCO (BATCHDIR) +\+MyFile.pdf", "@APPATH (export) +\+MyFile.pdf", true)</pre>	Moves the PDF file from one directory to another.

Parent topic: [FileIO actions](#)

SetFileReadOnly

Sets or removes the read only attribute from a file or set of files.

Member of namespace

FileIO

Syntax

```
bool SetFileReadOnly (bool readonly, string filename)
```

Parameters

readonly

Type: bool

filename

Type: string

Parameters

- readonly : true turns on the read only attribute, false turns it off.
- filename : The path and filename. Smart parameters and DOS wildcards are allowed.

Returns

Always True.

Level

All levels.

Details

Sets the read only attribute of a file, or a group of files. The read only attribute can be set or cleared with this action. Standard DOS * and ? wildcards can be used to affect a change on multiple files at once.

Example

```
SetFileReadOnly(false, "C:\MyDir\*.*.*)
```

This example sets all of the files in MyDir as read-write.

Parent topic: [FileIO actions](#)

SetProfileString

Writes a value to a profile file, typically called an INI file.

Member of namespace

FileIO

Syntax

```
bool SetProfileString (string filename, string section, string key, string value)
```

Parameters

filename
Type: string

section
Type: string

key
Type: string

value
Type: string

Parameters

- filename : The profile file name.
- section : The section within the file.
- key : specifies if the destination file should be overwritten.
- value : The value to write. Smart parameters are supported for each of the input parameters.

Returns

True if the value is written to the profile. Otherwise, False.

Level

All levels.

Details

Writes a value to a settings profile, typically a file with an INI extension. The value is stored in the variable provided as a parameter. If the variable does not exist, it will be created. If the variable does exist, the current value will be replaced with the new value. If the file does not exist, the file will be created.

If you are writing Unicode characters to the INI file, it is required that the file exist and that it is in Unicode format. If the file is not a Unicode format, the Unicode characters may appear incorrectly when read by `GetProfileString` or displayed in an editor. Most file editors will let you chose the type of file encoding. See the help for your specific editor.

Example

```
SetProfileString("C:\MyDir\config.ini", "MySection", "MyKey", "Batch+@BATCHID")
```

Parent topic: [FileIO actions](#)

SplitFileName

Splits a file name into user specified variables.

Member of namespace

FileIO

Syntax

```
bool SplitFileName (string inputFilename, string rootPathVariable, string pathVariable, string fileVariable, string extVariable)
```

Parameters

inputFilename
 Type: string
rootPathVariable
 Type: string
pathVariable
 Type: string
fileVariable
 Type: string
extVariable
 Type: string

Parameters

- inputFilename : The input file name to break into its three logical parts.
- rootPathVariable : The name of the variable to store the root path.
- pathVariable : The name of the variable to store the file path.
- fileVariable : The name of the variable to store the file name without the extension.
- extVariable : The name of the variable to store the file extension.

Smart parameters are supported for each of the input parameters. For each of the parameters that accept a variable, the level must be specified with a Smart Parameter.

It is not necessary to specify a variable name for each part of the path that is split. Only destination variables for the desired values need to be specified. If you do not want to store a particular file name part, leave that parameter blank.

Returns

False if the structure of the file name or path is invalid. Otherwise, True. The file does not need to exist for this action to succeed.

Level

All levels.

Details

This action splits the input file name into its root path, path, file name and the extension, and place each of the parts into their own variables. The file does not need to exist.

If the file contains a file extension, the saved file extension will contain a preceding period.

Example

Code	Comment
<pre>SplitFileName("C:\Dir1\Dir2\MyFile.abc", " @B.FROOT", "@B.FPATH", " @B.FNAME", "@B.FEXT")</pre>	<p>Splits the file name and saves the information into four variables at the batch level:</p> <ul style="list-style-type: none">• FROOT = "C:\"• FPATH = "C:\Dir1\Dir2"• FNAME = "MyFile"• FEXT = ".abc"
<pre>SplitFileName("@VAR(IMAGEFILE)", "", "", "", " @B.EXT")</pre>	<p>Obtains the value that is stored in the variable IMAGEFILE and saves the file extension into a variable EXT.</p>
<pre>SplitFileName("@P.ScanSrcPath", "", "", "@P\ Fieldname.Text", "")</pre>	<p>Obtains the value that is stored in the page variable ScanSrcPath and saves the file name into the subfield Text.</p>

Parent topic: [FileIO actions](#)

ZipOcrResults

Creates a .zip file to store recognition results for every page of the document.

Member of namespace

FileIO

Syntax

Bool ZipOcrResults ()

Parameters

None

Returns

True if the .zip file is created successfully, otherwise False.

Level

Document level.

Details

This action creates a .zip file that contains all the files with recognition results from every page of the document. The name of the .zip file is based on the ID of the document object, and is in the DocumentID.ocr.zip format.

Example

If an action is called on a document with an ID 20160914.000000.01, which contains two pages, then a .zip file that is named as 20160914.000000.01.ocr.zip gets created. This .zip file contains files that store recognition results that are associated with each page. For example, -

tm000001.xml

tm000001_layout.xml

tm000001c.xml

tm000002.xml

tm000002_layout.xml

tm000002c.xml

Parent topic: [FileIO actions](#)

FileNetIDM actions

Use the FileNetIDM actions to upload documents into an FileNet® Image Services library

The FileNetIDM actions integrate Datacap applications with the FileNet Image Services library. You run these actions to access the FileNet Image Services server, set up document attributes and folders on the server, and upload documents to the server for storage.

- [AddAllImagesToDocument](#)
Adds all Document Page object images to the Image Services document object.
- [AddFileToDocument](#)
Adds a file to the current FileNet document.

- [AddPDFImageToDocument](#)
Adds a PDF image file to the new FileNet document.
- [AddTIFImageToDocument](#)
Adds a TIF image file to the new FileNet document.
- [CreateFolder](#)
Creates a top-level FileNet folder.
- [FileNetDB_ADOConnect](#)
Establishes an ActiveX Data connection object with FileNet.
- [FileNETDocID_SaveAsSmartParameter](#)
Assigns the ID of the FileNet document to a variable of the bound object in the Document Hierarchy.
- [FileNETDocID_SetValue](#)
Assigns the ID of the FileNet document to a child object of the bound object in the Document Hierarchy.
- [GetDocuments](#)
Logs the names of the documents in the FileNet collection.
- [GetTopFolders](#)
Lists the existing top-level folders in the log file of the task.
- [IndexProperty_ID_Component](#)
Links the property of the FileNet document to an object of the Document Hierarchy.
- [IndexProperty_ID_DateComponent](#)
Sets up the Date element of the processing index of the FileNet task.
- [IndexProperty_ID_Value](#)
Assigns a constant value to a particular property of a FileNet document.
- [IndexProperty_LeftJUSTIFY](#)
Left-justifies a value that is being assigned to a target property of the FileNet document.
- [IndexProperty_RightJUSTIFY](#)
Right-justifies a value that is being assigned to a target property of the FileNet document.
- [IndexProperty_SmartParameter](#)
Assigns a constant value to a particular property of a FileNet document.
- [Library_DMA_Initialize](#)
Initializes but does not open the FileNet DMA library.
- [Library_DS_Initialize](#)
Initializes a previously defined, active Document Services Library.
- [Library_IS_Initialize](#)
Initializes a previously defined, active Image Services library.
- [Library_LogIn](#)
Logs in to the Image Services library by using the user ID and Password parameter values.
- [Library_LogOff](#)
Closes the FileNet connection to the Image Services library.
- [NewDocument](#)
Sets up a new document and assigns a previously defined Document Class to it.
- [SaveDocToFolder](#)
Saves the document into an existing folder of the open FileNet library.
- [Upload](#)
Uploads the active FileNet document to the open FileNet library.
- [Upload_SetDelay](#)
Controls the delay between upload retries.
- [Upload_SetNumAttempts](#)
Sets the number of attempts to complete a failed Upload action.
- [UseIndexes_OFF](#)
Turns off the Rulerunner indexing feature.
- [UseIndexes_ON](#)
Allows the task to access the properties of the FileNet document and provide these values to the objects in the Document Hierarchy.

Parent topic: [Global actions](#)

AddAllImagesToDocument

Adds all Document Page object images to the Image Services document object.

Syntax

```
bool AddAllImagesToDocument ()
```

Parameters

None.

Returns

False if action is placed at the Batch level, if the current active FileNet® document has already been committed to the Library, or if no documents exist in the batch. Otherwise, True.

Attention: If the action cannot access the batch's image files, the action directs the Rulerunner task to finish with a status of Aborted.

Level

Document, Page or Field levels.

Details

Assigns all images associated within a Document object (or parent Document object) of the Document Hierarchy to the new FileNet document.

This action is valid for IS libraries only. DS libraries only permit a single associated file. This action solicits information from the Rulerunner task's Page file (upload.xml, for example) as it assigns Image files representing pages linked to a Document object or child Page objects of the Document Hierarchy to the FileNet document.

The images are not yet committed to the library.

Example

```
NewDocument ("1040EZtwo")  
AddAllImagesToDocument ()
```

Parent topic: [FileNetIDM actions](#)

AddFileToDocument

Adds a file to the current FileNet® document.

Syntax

```
bool AddFileToDocument (StrParamMW)
```

Parameters

String value of the File name to add to the document and its path. Smart Parameters are supported.

Returns

False if the specified file is not found, or if the active FileNet document has already been committed to the library. Otherwise, True.

Attention: If the action cannot access the batch's image files, the action directs the Rulerunner task to finish with a status of Aborted.

Level

All levels.

Details

Adds any file you designate as a parameter to the current FileNet document.

If the parameter does not include a path to a folder, the action will use the path to the current Batches directory as the default. You can also designate a variable of the bound object of the Document Hierarchy as the source of path's value by using the # character followed by the variable's name. For example: *#FilePath*.

Example

```
NewDocument ("1040EZtwo")
AddTIFFImageToDocument ()
AddFileToDocument ("C:\Datacap\MQSW\Process\FNLog.log")
```

This sequence assumes that Datacap logs its FileNet activities and that a resulting Log file is available for the document.

Parent topic: [FileNetIDM actions](#)

AddPDFImageToDocument

Adds a PDF image file to the new FileNet® document.

Syntax

```
bool AddPDFImageToDocument ()
```

Parameters

None.

Returns

False if no Page component of the calling object is found, batch images cannot be accessed, or if the active FileNet document has already been committed to the library. Otherwise, True.

Attention: If the action cannot access the batch's image files, the action directs the Rulerunner task to finish with a status of Aborted.

Level

Field or Page level.

Details

Adds a PDF Image file associated with a Page object of the Document Hierarchy to the new FileNet document. This action works only if the PDF file is in the appropriate folder of the application's Batches directory – and has the same name as an associated page's corresponding Tiff Image file.

Example

```
NewDocument ("1040EZtwo")  
AddPDFImageToDocument ()
```

This sequence associates calling components bound to the PDF file with a new FileNet '1040EZtwo' document.

Parent topic: [FileNetIDM actions](#)

AddTIFImageToDocument

Adds a TIF image file to the new FileNet® document.

Syntax

```
bool AddTIFImageToDocument ()
```

Parameters

None.

Returns

False, if no Page component of the calling object is found, batch images cannot be accessed, or if the active FileNet document is already committed to the library. Otherwise, True.

Attention: If the action cannot access the batch's image files, the action directs the Rulerunner task to finish with a status of Aborted.

Level

Field or Page level.

Details

Adds the Image file that is associated with a Page object of the Document Hierarchy to the new FileNet document.

If a rule that contains this action is bound to a Field object, it adds the Image file that is associated with the field's parent Page object. An Upload action must stay at any level lower than this action.

Example

```
NewDocument ("1040EZtwo")  
AddTIFImageToDocument ()
```

This sequence associates the calling component's bound Image file with a new FileNet 1040EZtwo document.

Parent topic: [FileNetIDM actions](#)

CreateFolder

Creates a top-level FileNet® folder.

Syntax

```
bool CreateFolder (StrParamMW)
```

Parameters

The name of the folder to create. Smart Parameters are supported.

Returns

False if the folder cannot be created. Otherwise, True.

Level

Document, Page or Field levels.

Details

Creates a top-level FileNet folder.

Example

```
CreateFolder("MQSW_Q601")
```

Parent topic: [FileNetIDM actions](#)

FileNetDB_ADOConnect

Establishes an ActiveX Data connection object with FileNet®.

Syntax

```
bool FileNetDB_ADOConnect ()
```

Parameters

None.

Returns

False if the FileNet library is not initialized. Otherwise, True.

Attention: If the action encounters an error connecting to the specified database, the action directs the Rulerunner task to finish with a status of Aborted.

Level

All levels.

Details

Establishes an ActiveX Data Connection object (ADO connection) with the specified FileNet database.

Example

```
Library_IS_Initialize("DefaultIMS:Domain:FileNet")
LibraryLogin("Admin,AdOK")
FileNetDB_ADOConnect()
```

Parent topic: [FileNetIDM actions](#)

FileNETDocID_SaveAsSmartParameter

Assigns the ID of the FileNet® document to a variable of the bound object in the Document Hierarchy.

Syntax

```
bool FileNETDocID_SaveAsSmartParameter ()
```

Parameters

None.

Returns

False, if no active FileNet document is found or if the active FileNet document has not been committed.
Otherwise, True.

Level

All levels.

Details

Assigns the FileNet document's ID to the TEXT variable of the bound object of the Document Hierarchy.

Example

```
NewDocument("1040EZtwo")
AddAllImagesToDocument()
Upload()
FileNETDocID_SaveAsSmartParameter()
```

Parent topic: [FileNetIDM actions](#)

FileNETDocID_SetValue

Assigns the ID of the FileNet® document to a child object of the bound object in the Document Hierarchy.

Syntax

```
bool FileNETDocID_SetValue (StrParamMW)
```

Parameters

The name of the Field object. Smart Parameters are supported.

Returns

False if an active FileNet document is not found; if the active FileNet document has not been committed; or if the child Field object was not found. Otherwise, True.

Level

Document level, Page level or Field with child fields.

Details

Assigns the FileNet Document's ID to a child Field object of the bound Document, Page or parent Field object of the Document Hierarchy.

Example

```
NewDocument ("1040EZtwo")  
AddAllImagesToDocument ()  
Upload ()  
FileNETDocID_SetValue ("DocID")
```

This sequence will set up a new FileNet document, commit it to the FileNet library, and assign its ID as the Text value of the specified child field value of the bound object.

Parent topic: [FileNetIDM actions](#)

GetDocuments

Logs the names of the documents in the FileNet® collection.

Syntax

```
bool GetDocuments ()
```

Parameters

None.

Returns

Always True.

Level

All levels.

Details

A utility action used to aid debugging, it logs the names of the documents in the collection. This action is used to help verify the FileNet connection and that documents have been created. It is recommended that this is used for debugging and not in a normal production environment.

Example

```
GetDocuments ()
```

Parent topic: [FileNetIDM actions](#)

GetTopFolders

Lists the existing top-level folders in the log file of the task.

Syntax

```
bool GetTopFolders ()
```

Parameters

None.

Returns

False if the top-level folder collection cannot be located. Otherwise, True.

Level

All levels.

Details

Lists existing top-level FileNet® folders in the current task's Log file.

Example

```
GetTopFolders ()
```

Parent topic: [FileNetIDM actions](#)

IndexProperty_ID_Component

Links the property of the FileNet® document to an object of the Document Hierarchy.

Syntax

```
()
```

Parameters

Comma-separated values of:

1. The name of a FileNet document's property;

2. The name(s) of one or more Document Hierarchy objects with values for the variable that will be transferred to the FileNet document's property;
3. The value of the maximum length of the subscript value.

Smart Parameters are supported.

Returns

False if the FileNet property specified is invalid, or the FileNet Property Collection does not exist. Otherwise, True.

Attention: If the action returns False, the action directs the Rulerunner task to finish with a status of Aborted.

Level

Page level or Field level with child Fields.

Details

This action links the property of the FileNet document to an object of the Document Hierarchy.

Example

```
IndexProperty_ID_Component("FNBatch,1040EZ,18")
IndexProperty_ID_Component("FNDoc,1040EZTwo,22")
IndexProperty_ID_Component("FNPageF,Front,12")
IndexProperty_ID_Component("FNPageB,Back,12")
IndexProperty_ID_Component("FNfldLast,Last,18")
IndexProperty_ID_Component("NFldFirst,First,18")
IndexProperty_ID_Component("NFldMI,MI,2")
```

This sequence defines seven elements of a processing index for a FileNet task. In each case, the first parameter is the name of a property of a FileNet document that has been previously assembled. The second parameter assigns the Name of a Document Hierarchy object to the FileNet document's property. The third parameter is the property's maximum length.

During task operations, runtime values for each object will become the FileNet document's Index values.

Parent topic: [FileNetIDM actions](#)

IndexProperty_ID_DateComponent

Sets up the Date element of the processing index of the FileNet® task.

Syntax

```
bool IndexProperty_ID_DateComponent (StrParamMW)
```

Parameters

String consisting of four comma-separated values:

1. The name of the FileNet document's Date property
2. The name of a Document Hierarchy object with a Date property
3. The format of the Date when supplied to the FileNet document

4. The format of the Date value that is being added to the task's processing index. See the example for acceptable Date values.

Smart Parameters are supported.

Returns

False if the FileNet property specified is invalid, or the FileNet Property Collection does not exist. Otherwise, True.

Attention: If the action returns False, the action directs the Rulerunner task to finish with a status of Aborted.

Level

Page level or Field level with child Fields.

Details

This element can supply Date information to a Date property of the FileNet document.

The parameters can use these Date formats: cc Century 20 yy Year 03 yyyy Year 2003 dd Day 29 mm Month 06 Julian Year/Day 03-145

Example

```
IndexProperty_ID_Date_Component ("FNStart,1040EZ,mm/dd/yy,yyyy/mm/dd")
```

This example re-formats the value of the Batch object's Start Date and assigns it to the FNStart property of the FileNet document.

Parent topic: [FileNetIDM actions](#)

IndexProperty_ID_Value

Assigns a constant value to a particular property of a FileNet® document.

Syntax

()

Parameters

String with two comma-separated values:

1. The name of the FileNet document's target property;
2. The constant's value.

Smart Parameters are supported.

Returns

False, if the FileNet property specified cannot be set. Otherwise, True.

Level

All levels.

Details

Assigns a constant value to a particular property of a FileNet document.

Example

The following action assigns “3” to the FNTaxQtr property of the FileNet document when the FileNet task processes the bound object of the Document Hierarchy, the object to which the action’s rule applies.

```
IndexProperty_ID_Value (FNTaxQtr, 3)
```

Parent topic: [FileNetIDM actions](#)

IndexProperty_LeftJUSTIFY

Left-justifies a value that is being assigned to a target property of the FileNet® document.

Syntax

```
bool IndexProperty_LeftJUSTIFY (StrParamMW)
```

Parameters

String with two comma-separated values:

1. The name of the property of the FileNet document; and
2. The maximum size of a value.

Smart Parameters are supported.

Returns

False if the FileNet property cannot be set. Otherwise, True.

Level

All levels.

Details

Left-justifies a value that is being assigned to a target property of the FileNet document.

Example

```
IndexProperty_ID_Variable ("FNFldData, Year+SSN+ Income+Deductions+Net, 256")  
IndexProperty_LeftJUSTIFY ("FNFldData, 256")
```

This sequence provides the FNFldData property with a value, then formats the value before it is actually assigned to the active FileNet document.

Parent topic: [FileNetIDM actions](#)

IndexProperty_RightJUSTIFY

Right-justifies a value that is being assigned to a target property of the FileNet® document.

Syntax

```
bool IndexProperty_RightJUSTIFY (StrParamMW)
```

Parameters

String with two comma-separated values:

1. The name of the property of the FileNet document; and
2. The maximum size of a value.

Smart Parameters are supported.

Returns

False if the FileNet property cannot be set. Otherwise, True.

Level

All levels.

Details

Right-justifies a value that is being assigned to a target property of the FileNet document.

Example

```
IndexProperty_ID_Variable("FNFldData,Year+SSN+ Income+Deductions+Net,256")  
IndexProperty_RightJUSTIFY("FNFldData,256")
```

This sequence provides the FNFldData property with a value, then formats the value before it is actually assigned to the active FileNet document.

Parent topic: [FileNetIDM actions](#)

IndexProperty_SmartParameter

Assigns a constant value to a particular property of a FileNet® document.

Syntax

```
bool IndexProperty_SmartParameter (string PropertyName, string Value, string Length)
```

Parameters

1. The name of the FileNet document's target property.
2. The value to assign to the property.
3. The length of the property value (space filled).

Smart Parameters are supported for all arguments.

Returns

False, if the FileNet property specified cannot be set. Otherwise, True.

Level

All levels.

Details

Assigns a constant value to a particular property of a FileNet document.

Example

The following action assigns the value of field Taxes to the FNTaxQtr property of the FileNet document.

```
IndexProperty_SmartParameter (FNTaxQtr, @P/Taxes, "")
```

The following action assigns the space filled value of page variable *LastName* to the FNNameLast property of the FileNet document, if the variable value exceeds 10 characters the value that is saved is right-truncated.

```
IndexProperty_SmartParameter (FNNameLast, @P.LastName, 10)
```

Parent topic: [FileNetIDM actions](#)

Library_DMA_Initialize

Initializes but does not open the FileNet® DMA library.

Syntax

```
bool Library_DMA_Initialize (StrParam)
```

Parameters

String value consisting of three colon-separated elements of the Library Name. Smart Parameters are supported.

Returns

False if there is a problem connecting to the FileNet DMA Library. Otherwise, True.

Note: If the action returns False, the action directs the Rulerunner task to finish with a status of Aborted.

Level

Any level, but the Batch level is recommended.

Details

Do not confuse the Library Name with the local FileNet Neighborhood label. In some cases, the formal three-part Library Name must be used to properly configure initialization. Please check your FileNet documentation

on how to discern your formal library name.

Example

```
Library_DMA_Initialize("DMALibrary:Datacap:FileNet")
```

This action initializes but does not open the library – see [Library_Login](#).

Parent topic: [FileNetIDM actions](#)

Library_DS_Initialize

Initializes a previously defined, active Document Services Library.

Syntax

```
bool Library_DS_Initialize (StrParam)
```

Parameters

String value consisting of the three colon-separated elements of the Library Name. Smart Parameters are supported.

Returns

False if there is a problem connecting to the FileNet® Document Services Library. Otherwise, **True**.

Note: If the action returns False, the action directs the Rulerunner task to finish with a status of Aborted.

Level

Any level, but the Batch level is recommended.

Details

Do not confuse the Library Name with the local FileNet Neighborhood label. In some cases, the formal three-part Library Name must be used to properly configure initialization. Please check your FileNet documentation for guidelines on designating a formal Library Name.

Example

```
Library_DS_Initialize("DSLlibrary:Datacap:FileNet")
```

This action initializes but does not open the library – see [Library_Login](#).

Parent topic: [FileNetIDM actions](#)

Library_IS_Initialize

Initializes a previously defined, active Image Services library.

Syntax

```
bool Library_IS_Initialize (StrParam)
```

Parameters

String value containing the Library name. The Library name will commonly consist of three, colon-separated elements of the Library Name. In some cases, the short name can be used. Smart Parameters are supported.

Returns

False if there is a problem connecting to the FileNet® Image Services Library. Otherwise, True.
Attention: If the action returns False, the action directs the Rulerunner task to finish with a status of Aborted.

Level

Any level, but the Batch level is recommended.

Details

Initializes a previously defined, active Image Services Library.

In some cases, the formal three-part Library Name should be used to properly configure initialization. However, there may be some cases where configuring the library through the IDM Configuration tool first and then passing the short name could be used. Please check your FileNet documentation for guidelines on designating a formal Library Name.

Example

```
Library_IS_Initialize("LibraryName")
```

This action initializes but does not open the library – see [Library_Login](#).

Parent topic: [FileNetIDM actions](#)

Library_Login

Logs in to the Image Services library by using the user ID and Password parameter values.

Syntax

```
bool Library_Login (StrParam)
```

Parameters

String values of the User ID and Password, with a comma separator. Smart Parameters are supported.

Returns

False if an active library is not found, the parameter values are incorrect, or an error occurs while logging into the library. Otherwise, True.
Attention: If the action returns False, the action directs the Rulerunner task to finish with a status of Aborted.

Level

Any level, but the Batch level is recommended.

Details

Logs into the initialized FileNet® library using the User ID and Password parameter values. You must include this action to access a library. Be sure the Library_Login action follows one of the Library_Initialize actions.

Example

```
Library_DS_Initialize("DefaultLib:Datacap:FileNet")
Library_Login("FileNet2, FN2")

Library_IS_Initialize("LibraryName")
Library_Login("@APPVAR(values/gen/ISUser), @APPVAR
(values/adv/ISPassword) ")

Library_IS_Initialize("LibraryName")
Library_Login("@STRING(@APPVAR
(values/gen/ISUser), @APPVAR(values/adv/ISPassword) ")")
```

Parent topic: [FileNetIDM actions](#)

Library_LogOff

Closes the FileNet® connection to the Image Services library.

Syntax

```
bool Library_LogOff ()
```

Parameters

None.

Returns

True, if the logoff was successful. Otherwise, False.

Level

Any level, but the Batch level is recommended.

Details

Closes the FileNet connection.

Example

```
Library_LogOff()
```

Parent topic: [FileNetIDM actions](#)

NewDocument

Sets up a new document and assigns a previously defined Document Class to it.

Syntax

```
bool NewDocument (string StrParam)
```

Parameters

String value of a previously defined FileNet® Document Class. Smart Parameters are supported.

Returns

False if a new document cannot be created. Otherwise, True.

Attention: If the action returns False, the action directs the Rulerunner task to finish with a status of Aborted.

Level

All Levels.

Details

Sets up a new FileNet document, and assigns the FileNet Document Class you specify as an argument to the new FileNet document.

After an Upload action commits the document to a FileNet library, FileNet immediately links a unique ID to the document.

Example

```
Library_DS_Initialize("DefaultIMS:Datacap:FileNet")
Library_Login("FileNet2, FN2")
NewDocument("1040EZtwo")
```

In this example, the NewDocument action instantiates a new FileNet document of Class:'1040EZtwo'. To populate the document, you'll probably follow this action with one of the AddImage actions.

Parent topic: [FileNetIDM actions](#)

SaveDocToFolder

Saves the document into an existing folder of the open FileNet® library.

Syntax

```
bool SaveDocToFolder (StrParam)
```

Parameters

String value of the Folder ID, beginning with a forward slash (/) - see the example. Smart Parameters are supported.

Returns

False if there is no active FileNet document, no active FileNet library, invalid parameter, or if the active FileNet document has not been committed. Otherwise, True.

Attention: If the action cannot access the specified folder, the action directs the Rulerunner task to finish with a status of Aborted.

Level

All Levels.

Details

Places the committed FileNet document in an existing folder of the open FileNet library. Although the forward slash (/) character is a standard element of this action's parameter, the setup of your FileNet library may mean that the forward slash is not used. Under exceptional circumstances, this action will have this syntax – note that a forward slash does not precede the Folder ID: `SaveDocToFolder("1074a")`.

Example

```
Library_DS_Initialize("1040Docs")
Library_Login("FileNet2, FN2")
NewDocument("1040EZtwo")
AddAllImagesToDocument()
Upload()
SaveDocToFolder("/1074a")
```

As the example shows, you can insert this action after adding images and successfully committing (uploading) the document to the FileNet library.

Parent topic: [FileNetIDM actions](#)

Upload

Uploads the active FileNet® document to the open FileNet library.

Syntax

```
bool Upload ()
```

Parameters

None.

Returns

False if the Document object does not exist; if the library object is missing; if all pages were previously committed; if the active FileNet document has already been committed to the library; or the upload is unsuccessful. Otherwise, True.

Attention: If the active FileNet document has already been committed, or the action encounters an error, the action directs the Rulerunner task to finish with a status of Aborted.

Level

All levels.

Details

Commits the active FileNet document to the open FileNet library.

Example

```
Upload()
```

Parent topic: [FileNetIDM actions](#)

Upload_SetDelay

Controls the delay between upload retries.

Member of namespace

FileNetIDM

Syntax

```
bool Upload_SetDelay (StrParamMW)
```

Parameters

The number of seconds to wait between each retry of the FileNet upload. Smart Parameters are supported

Returns

Always True.

Level

All levels.

Details

If an upload fails, the application will retry automatically to upload. Between each retry, the application will pause. You can use this action to set the length of the pause of the application. If this action is not called, the default value of 5 seconds is used.

Example:

```
Upload_SetDelay("15")
```

Parent topic: [FileNetIDM actions](#)

Upload_SetNumAttempts

Sets the number of attempts to complete a failed Upload action.

Syntax

```
bool Upload_SetNumAttempts (StrParamMW)
```

Parameters

The number of times to retry the FileNet upload upon failure. Smart Parameters are supported.

Returns

Always True.

Level

All levels.

Details

If the upload action fails, it will be automatically retried. The number of retries can be controlled with this action. If this action is not called, the default value of 3 will be used.

Example

```
Upload_SetNumAttempts ("5")
```

Parent topic: [FileNetIDM actions](#)

UseIndexes_OFF

Turns off the Rulerunner indexing feature.

Syntax

```
bool UseIndexes_OFF ()
```

Parameters

None.

Returns

Always True.

Level

All levels.

Details

Turns Off a Rulerunner task's Indexing feature. Because the feature is On by default, the task will continue to generate and assign index values until a rule with this action turns it Off.

A rule with action must be applied before a new FileNet® document is created.

Example

```
UserIndexes_OFF ()
```

Parent topic: [FileNetIDM actions](#)

UseIndexes_ON

Allows the task to access the properties of the FileNet® document and provide these values to the objects in the Document Hierarchy.

Syntax

```
bool UseIndexes_ON ()
```

Parameters

None.

Returns

Always True.

Level

All levels.

Details

This status allows the task to access the properties of the FileNet document, and to provide these properties with values of objects of the Document Hierarchy. True is the default value for using indexes.

Example

```
UserIndexes_ON()  
IndexProperty_ID_Component ("FNDoc,1040EZTwo,12")
```

A task cannot define or populate indices until a rule with this action activates the Indexing feature. However, the On status is a default status, and is in effect unless a UseIndexes_OFF action turns it Off.

Parent topic: [FileNetIDM actions](#)

FileNet P8 actions

Use the FileNet® P8 actions to export data to a FileNet P8 repository.

The FileNet P8 actions integrate Datacap applications with the IBM® FileNet P8 repository. You run these actions to access the FileNet P8 server, set up document attributes and folders on the server, and upload documents to the server for storage.

- [FNP8_AddRedactionsToP8Document](#)
Updates an existing FileNet P8 document to add redactions, if any of the fields have been redacted.
- [FNP8_CreateFolder](#)
Creates a subfolder on a specified target class and object.
- [FNP8_Login](#)
Sets the user ID and password for login to the FileNet P8 system.
- [FNP8_MultiPageDocs](#)
Sets the upload mode to create a multiple page FileNet P8 document.

- [FNP8_SearchAndDownload](#)
Executes a search for documents in FileNet P8 and downloads the content of the documents.
- [FNP8_SetSearchAndDownloadStatusProperty](#)
Sets the name of an integer property that the FNP8_SearchAndDownload action updates for the documents that are downloaded. This indicates that the document has been downloaded.
- [FNP8_SetSearchClass](#)
Sets the document class to be used for search.
- [FNP8_SetSearchCurrentVersionOnly](#)
Set to true to include only the current version of documents in the search results, false otherwise. Default is true.
- [FNP8_SetSearchDownloadDir](#)
Sets the directory path to the location at which the contents of the documents that are returned by search are downloaded.
- [FNP8_SetSearchFolderRestriction](#)
Sets a folder path to restrict the search results.
- [FNP8_SetSearchIncludeDocsWithoutContent](#)
Set to "True" to include documents without content in search results. By default it is set to "False".
- [FNP8_SetSearchIncludeSubClasses](#)
Set to "True" to include the subclasses of the search class in search results. Otherwise, "False". Default is "True".
- [FNP8_SetSearchMaxItems](#)
Sets the maximum number of search results.
- [FNP8_SetSearchOrderBy](#)
Sets the ORDERBY clause for the search results.
- [FNP8_SetSearchWhereClause](#)
Sets the search criteria (WHERE clause).
- [FNP8_SetDefineSecurityParentage](#)
Specifies whether the scanned documents that are exported from Datacap to the FileNet Content Manager inherit security from the parent folder.
- [FNP8_SetDestinationFolder](#)
Sets the destination folder for the documents to be uploaded.
- [FNP8_SetDocClassId](#)
Sets the FileNet P8 document class for the uploaded files.
- [FNP8_SetDocTitle](#)
Sets the document title for documents that you are uploading.
- [FNP8_SetFileMimeType](#)
Associates a custom MIME type with a file extension for documents that are uploaded to the FileNet P8 system.
- [FNP8_SetFileType](#)
Assigns the file type for the files that are uploaded.
- [FNP8_SetKeyProperty](#)
Sets the update key to a FileNet document property and its corresponding property value.
- [FNP8_SetLocale](#)
Identifies the locale on the target FileNet P8 system.
- [FNP8_SetMultiValueProperty](#)
Sets the values in a multi-value property.
- [FNP8_SetProperty](#)
Sets the designated FileNet P8 property to a specified value.
- [FNP8_SetPropertyEx](#)
Sets the designated FileNet P8 property to a specified value.
- [FNP8_SetRetry](#)
Sets the number of automatic upload retries.

- [FNP8_SetTargetClassID](#)
Sets the FileNet P8 document class for uploaded documents.
- [FNP8_SetTargetObjectID](#)
Sets the name of the Object Store in which documents are stored.
- [FNP8_SetTimeout](#)
Sets the timeout for the FileNet P8 web service in milliseconds.
- [FNP8_SetUploadMode](#)
Sets the upload mode.
- [FNP8_SetURL](#)
Sets the URL for the FileNet P8 Server that is used.
- [FNP8_UpdateProperties](#)
Updates the properties of an existing FileNet P8 document by using the data that is passed into the FNP8_SetProperty action.
- [FNP8_Upload](#)
Uploads the batch images to the FileNet P8 repository.
- [FNP8_UploadDir](#)
Uploads all of the images in the folder to the specified destination folder.

Parent topic: [Global actions](#)

FNP8_AddRedactionsToP8Document

Updates an existing FileNet P8 document to add redactions, if any of the fields have been redacted.

Syntax

```
bool FNP8_AddRedactionsToP8Document ()
```

Parameters

None. The page property "PageName" is used.

Returns

False if any issues occur during addition of redactions. Otherwise, True.

Level

Page level.

Details

This action updates an existing FileNet P8 document by adding redactions (if any of the fields have been redacted).

Important: This action can be used only for documents that were downloaded by FNP8_SearchAndDownload action, or those documents that have file names that begin with the FileNet P8 document ID. An example of the file name format is "{3054025F-0000-CE12-B41A-7B2661D28802}.APT008.tif".

Example

```
FNP8_SetSearchClass ("Invoice")  
FNP8_SetSearchDownloadDir ("C:/mylocaldir")  
FNP8_SetSearchWhereClause ("DocumentTitle LIKE '2017-Invoices%'  
FNP8_SearchAndDownload ()
```

```
Scan()  
RedactFields("CCPattern1", "Redact_CreditCard", "4,4,4,4", True)  
FNP8_AddRedactionsToP8Document()
```

Parent topic: [FileNet P8 actions](#)

FNP8_CreateFolder

Creates a subfolder on a specified target class and object.

Syntax

```
bool FNP8_CreateFolder(StrParam)
```

Parameters

A string value or a predefined Smart Parameter variable which specifies the name of the folder to create.

The allowed predefined variables are: *@BATCHID*, *@ID*, *@STATUS*, *@TYPE*, *@VALUE*, *@JOBID*, *@JOBNAME*, *@OPERATOR*, *@STATION*, *@TASKID*, and *@TASKNAME*. Please refer to the smart parameter documentation for more information on these values.

Returns

False if the parameter cannot be parsed, the set up information is invalid, or the folder cannot be created. Otherwise, True.

Attention: If the action returns False, the action directs the Rulerunner task to finish with a status of Aborted.

Level

All levels.

Details

This action creates a subfolder on a specified target class and object. Like the Upload actions, this action must be preceded by SetURL, Login and SetTargetClassID actions.

Example

```
FNP8_SetDestinationFolder("/1040EZ/Export/")  
FNP8_CreateFolder("@BATCHID")
```

This example creates a subfolder under the \1040EZ\Export\ folder and change the destination folder to the newly created folder. If the folder is created successfully, the action adds a variable *Folder_ID* to the current DCO with the folder ID returned from FileNet® Web Service.

Parent topic: [FileNet P8 actions](#)

FNP8_Login

Sets the user ID and password for login to the FileNet® P8 system.

Syntax

```
bool FNP8_Login(StrParam)
```

Parameters

Two comma-separated smart parameter supported string values:

1. Login Name.
2. Password.

Returns

False, if DC_P8_Server.dll was not successfully installed on this computer, or either parameter value is missing or not a string. Otherwise, True.

Level

All, but generally at the Batch level.

Details

This action provides the User ID and Password to use when logging in to IBM® FileNet P8.

Example

```
FNP8_Login("User1,Password1")
```

Parent topic: [FileNet P8 actions](#)

FNP8_MultiPageDocs

Sets the upload mode to create a multiple page FileNet® P8 document.

Syntax

```
bool FNP8_MultiPageDocs(StrParam)
```

Parameters

Specifies whether the FileNet_MultiPageDocs action creates in IBM® FileNet Content Manager a single content element for all the Datacap document pages or one content element per Datacap document page.

Returns

False if the parameter is blank or if the value of the parameter is invalid. Otherwise, True.

Level

All levels

Details

Set FNP8_MultiPageDocs to True to configure the upload actions to create one FileNet P8 document that contains multiple pages.

If the `FNFP8_MultiPageDocs` action is set to `True`, the `FNFP8_Upload` or `FNFP8_UploadDir` actions place all of the pages that are listed the DCO object into a single FileNet P8 document. The default behavior for these actions is to create a FileNet P8 document for each individual page that is listed in the document DCO object.

Example:

```
FNFP8_MultiPageDocs ("True")
FNFP8_Upload()
```

Parent topic: [FileNet P8 actions](#)

FNFP8_SearchAndDownload

Executes a search for documents in FileNet P8 and downloads the content of the documents.

Syntax

```
bool FNFP8_SearchAndDownload()
```

Parameters

Uses the parameters set in the `FNFP8_SetSearch*` methods for executing the search against FileNet P8 repository.

Returns

False if the actions `FNFP8_SetURL`, `FNFP8_Login`, `FNFP8_SetTargetObjectID`, `FNFP8_SetSearchClass`, and `FNFP8_SetSearchDownloadDir` are not successfully called before this action. Otherwise, True.

Level

All levels.

Details

Executes a search for documents in FileNet P8 and downloads the content of the documents that are returned by the search.

By default, the maximum number of search results is limited to 1000 documents. Use the action `FNFP8_SetSearchMaxItems` to set a different maximum limit.

The user or process that executes the action must have 'write' permissions for the directory. If a file exists in the directory with the same name as the content that gets downloaded, the existing file gets overwritten.

Important: To prevent downloading duplicate item content, the task that calls `FNFP8_SearchAndDownload` action should not be run in a multi-threaded configuration. Instead, use single-threaded tasks. For information about configuring threads in Rulerunner, see [Configuring Rulerunner to run tasks](#).

The following actions are *required* to be called before you call `FNFP8_SearchAndDownload`.

- [FNFP8_SetURL](#)
- [FNFP8_Login](#)
- [FNFP8_SetTargetObjectID](#)
- [FNFP8_SetSearchClass](#)
- [FNFP8_SetSearchDownloadDir](#)

The following actions can *optionally* be called before you call FNP8_SearchAndDownload.

Important: Use the following actions to filter the documents that are to be downloaded by the FNP8_SearchAndDownload action.

- [FNP8_SetSearchWhereClause](#)
- [FNP8_SetSearchFolderRestriction](#)
- [FNP8_SetSearchOrderBy](#)
- [FNP8_SetSearchMaxItems](#)
- [FNP8_SetSearchCurrentVersionOnly](#)
- [FNP8_SetSearchIncludeDocsWithoutContent](#)
- [FNP8_SetSearchIncludeSubClasses](#)
- [FNP8_SetSearchAndDownloadStatusProperty](#)

Example 1

```
FNP8_SetURL("http://myp8server:9080/wsi/FNCEWS40MTOM/")
FNP8_Login("User1,Password1")
FNP8_SetTargetObjectID("AP_ObjectStore")
FNP8_SetSearchClass("Invoice")
FNP8_SetSearchDownloadDir("@APPPATH(vscanimagedir)")
FNP8_SetSearchWhereClause("DocumentTitle LIKE '2017-Invoices%')")
FNP8_SetSearchMaxItems("100")
FNP8_SetSearchOrderBy("InvoiceDate ASC")
FNP8_SearchAndDownload()
Scan()
```

In this example following tasks are performed -

- It searches for documents of class "Invoice" that have a property DocumentTitle, that matches the wildcard string '2017-Invoice%'.
- It sorts the results in ascending order based on the InvoiceDate property.
- It takes the top 100 results and downloads the contents of those documents to the directory specified in the Smart Parameter for the VScan image directory.

The output file name is a concatenation of the FileNet P8 item ID and the name of original content file uploaded (if it exists). If the original file name does not exist, then the default file extension would be .tif if the mime type of the content is image/tiff An example of the filename is {3054025F-0000-CE12-B41A-7B2661D28802}.Flight1.tif..

When the Scan() action runs, the action checks the populated image directory, and processes as usual.

Example 2

```
FNP8_SetURL("http://myp8server:9080/wsi/FNCEWS40MTOM/")
FNP8_Login("User1,Password1")
FNP8_SetTargetObjectID("AP_ObjectStore")
FNP8_SetSearchClass("Invoice")
FNP8_SetSearchDownloadDir("@APPPATH(vscanimagedir)")
FNP8_SetSearchWhereClause("DocumentTitle LIKE '2017-Invoices%')")
FNP8_SetSearchMaxItems("100")
FNP8_SetSearchAndDownloadStatusProperty("myStatusProp")
FNP8_SearchAndDownload()
Scan()
```

This example is similar to earlier one. However, the main differences are as follows -

1. For each document that is downloaded, the property myStatusProp of the document is set to a value of 1.
2. During document search, the documents that have myStatusProp=1 are excluded from the search.

FNP8_SetSearchAndDownloadStatusProperty

Sets the name of an integer property that the FNP8_SearchAndDownload action updates for the documents that are downloaded. This indicates that the document has been downloaded.

Syntax

```
bool FNP8_SetSearchAndDownloadStatusProperty (StrParam)
```

Parameters

A name (symbolic name) of an existing integer property of the FileNet P8 document class specified by the action FNP8_SetSearchClass. Smart parameters are supported. The name of the property is not validated until the FNP8_SearchAndDownload action is executed.

Important: The FNP8_SearchAndDownload action updates the value of this specified property.

Returns

False if parameter is not a string. Otherwise, True. The existence of the property on the document class is not verified until FNP8_SearchAndDownload is run.

Level

All levels.

Details

This action is optional before you call action FNP8_SearchAndDownload.

Sets the name of an integer property that the FNP8_SearchAndDownload action updates for the documents that are downloaded. This indicates that the document has been downloaded (status property value is set to "1"). On subsequent executions of FNP8_SearchAndDownload(), the P8 documents that have been downloaded are not downloaded again.

If no status property is set, then properties on the P8 document are not updated by the FNP8_SearchAndDownload action when a P8 document is downloaded. Besides, no status property is checked when searching for documents. To reset a status property that was set earlier, set it to empty string""

Example 1

```
FNPN8_SetSearchClass ("Invoice")
FNPN8_SetSearchDownloadDir ("C:/mylocaldir")
FNPN8_SetSearchAndDownloadStatusProperty ("myStatusProp")
FNPN8_SearchAndDownload ()
```

In this example, FNP8_SearchAndDownload() downloads P8 documents of class "Invoice" and sets the property myStatusProp to "1" for each document that is downloaded. When FNP8_SearchAndDownload() action performs its search, any document with myStatusProp=1 gets excluded from the search and is not downloaded.

Example 2

```
Example:  
FNP8_SetSearchClass ("Invoice")  
FNP8_SetSearchDownloadDir ("C:/mylocaldir")  
FNP8_SetSearchAndDownloadStatusProperty ("")  
FNP8_SearchAndDownload ()
```

In this example, `FNP8_SetSearchAndDownloadStatusProperty("")` resets any previously set value for the download status property name. `FNP8_SearchAndDownload()` downloads P8 documents of class "Invoice", but does not set a status property on documents that have been downloaded. When `FNP8_SearchAndDownload()` action performs its search, there is no check of a status property, so P8 documents that may have been downloaded are included.

Parent topic: [FileNet P8 actions](#)

FNP8_SetSearchClass

Sets the document class to be used for search.

Syntax

```
bool FNP8_SetSearchClass (StrParam)
```

Parameters

A string value of the name (symbolic name) of the class of document to be searched in FileNet P8. The name must be the symbolic name of a FileNet P8 class. Smart parameters are supported. The name of the class is not validated until the `FNP8_SearchAndDownload` action is executed..

Returns

False if parameter is not a string. Otherwise, True.

Level

All levels.

Details

This action is required before you call action `FNP8_SearchAndDownload`.

Set the name (symbolic name in P8) of the class of document to be searched in FileNet P8.

By default, subclasses of the specified class are included in the search results. See action [FNP8_SetSearchIncludeSubClasses](#) to know how to include subclasses in search.

Example

```
FNP8_SetSearchClass ("Invoice")  
FNP8_SetSearchDownloadDir ("C:/mylocaldir")  
FNP8_SearchAndDownload ()  
Scan ()
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetSearchCurrentVersionOnly

Set to true to include only the current version of documents in the search results, false otherwise. Default is true.

Syntax

```
bool FNP8_SetSearchCurrentVersionOnly (StrParam)
```

Parameters

A string representing a boolean value ("True" or "False") for whether or not to limit the search results to only the current version of documents. Smart parameters are supported.

Returns

True.

Level

All levels.

Details

This action is optional prior to calling action FNP8_SearchAndDownload.

Set to "True" to limit the search results to only the current version of documents.

Set to "False" to allow search results to include any version of documents.

If not set, the default is True.

If an invalid value is given for the parameter (i.e. not "True" or "False"), then the action has no effect on the setting and the setting stays the same value as it was prior to the action.

Example

```
FNP8_SetSearchClass ("Invoice")
FNP8_SetSearchDownloadDir ("C:/mylocaldir")
FNP8_SetSearchCurrentVersionOnly ("False")
FNP8_SearchAndDownload ()
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetSearchDownloadDir

Sets the directory path to the location at which the contents of the documents that are returned by search are downloaded.

Syntax

```
bool FNP8_SetSearchDownloadDir (StrParam)
```

Parameters

A string value for the directory path to the location at which the contents of the documents that are returned by search are downloaded. Smart parameters are supported.

Returns

False if parameter is not a string. Otherwise, True.

Level

All levels.

Details

This action is *required* to be called before you call action FNP8_SearchAndDownload.

Sets the directory path to the location at which the contents of the documents that are returned by search are downloaded.

The user or process that runs the FNP8_SearchAndDownload action must have write permission to the directory specified by this action. This action does not validate the existence of the directory.

Example 1

```
FNP8_SetSearchClass("Invoice")
FNP8_SetSearchDownloadDir("@APPPATH(vscanimagedir)")
FNP8_SearchAndDownload()
Scan()
```

In this example, the action is called to download the content of the search documents to the directory specified by the Smart Parameter for the VScan image directory.

Example 2

```
FNP8_SetSearchClass("Invoice")
FNP8_SetSearchDownloadDir("C:/mylocaldir")
FNP8_SearchAndDownload()
Scan()
```

In this example, action is called to download the content of the search documents to the directory C:/mylocaldir.

Parent topic: [FileNet P8 actions](#)

FNP8_SetSearchFolderRestriction

Sets a folder path to restrict the search results.

Syntax

```
bool FNP8_SetSearchFolderRestriction (StrParam)
```

Parameters

A string of the folder path in FileNet P8 repository to restrict search results. Smart parameters are supported. The folder path is not validated until the FNP8_SearchAndDownload action is executed.

Returns

False if parameter is not a string. Otherwise, True.

Level

All levels.

Details

This action is *optional* before you call action FNP8_SearchAndDownload.

Set a FileNet P8 folder path to restrict the search results. The search results are not included in the subfolders of the folder. Set to empty string "" to reset a previously set value.

Note: Setting the folder restriction might result in including only document versions with version state value "Released", in search result.

Example

```
FNp8_SetSearchClass("Invoice")
FNp8_SetSearchDownloadDir("C:/mylocaldir")
FNp8_SetSearchFolderRestriction("/MyFolder/MySubfolderA")
FNp8_SearchAndDownload()
```

Parent topic: [FileNet P8 actions](#)

FNp8_SetSearchIncludeDocsWithoutContent

Set to "True" to include documents without content in search results. By default it is set to "False".

Syntax

```
bool FNp8_SetSearchIncludeDocsWithoutContent (StrParam)
```

Parameters

A string that represents a boolean value ("True" or "False") to indicate whether or not to include documents without content in the search results. Smart parameters are supported.

Returns

True.

Level

All levels.

Details

This action is *optional* before you call action FNP8_SearchAndDownload.

Set to "True" to include documents without content in search results.

Set to "False" to filter out documents without content from search results.

If not set, the default is "False".

If an invalid value is given for the parameter (that is, not "True" or "False"), then the action has no effect on the setting. The setting retains the value that was set before the action was called.

It is not required to change the default setting, since there is no content to download from content-less documents. This action is only provided in case of any unexpected behavior with the search clause that is used to filter out content-less documents, and a need to turn off the filtering is triggered.

Example

```
FNp8_SetSearchClass ("Invoice")
FNp8_SetSearchDownloadDir ("C:/mylocaldir")
FNp8_SetSearchIncludeDocsWithoutContent ("True")
FNp8_SearchAndDownload ()
```

Parent topic: [FileNet P8 actions](#)

FNp8_SetSearchIncludeSubClasses

Set to "True" to include the subclasses of the search class in search results. Otherwise, "False". Default is "True".

Syntax

```
bool FNp8_SetSearchIncludeSubClasses (StrParam)
```

Parameters

A string that represents a boolean value ("True" or "False") to indicate whether or not to include documents of the subclasses of the search class in the search results. Smart parameters are supported.

Returns

True.

Level

All levels.

Details

This action is *optional* before you call action FNp8_SearchAndDownload.

Set to "True" to include documents of the class that is specified by the action FNp8_SetSearchClass and all the subclasses of the search class in search results.

Set to "False" to include only documents of the class that is specified by the action FNp8_SetSearchClass in search results.

If not set, the default is "True".

If an invalid value is set for the parameter (that is not "True" or "False"), then the action has no effect on the setting. The setting retains the value that was set before the action was called.

Example

```
FNp8_SetSearchClass("Invoice")
FNp8_SetSearchDownloadDir("C:/mylocaldir")
FNp8_SetSearchIncludeSubClasses("False")
FNp8_SearchAndDownload()
```

Parent topic: [FileNet P8 actions](#)

FNp8_SetSearchMaxItems

Sets the maximum number of search results.

Syntax

```
bool FNp8_SetSearchMaxItems (StrParam)
```

Parameters

A string that represents an integer value for the maximum number of items to return in search results. Smart parameters are supported.

Returns

False if parameter is not a string numerical value. Otherwise, True.

Level

All levels.

Details

This action is *optional* before you call action FNp8_SearchAndDownload.

You can set the maximum number of search results. Set to "0" or any value less than "0" for no maximum. If not set, the default maximum is 1000.

Example

```
FNp8_SetSearchClass("Invoice")
FNp8_SetSearchDownloadDir("C:/mylocaldir")
FNp8_SetSearchMaxItems("100")
FNp8_SearchAndDownload()
```

Parent topic: [FileNet P8 actions](#)

FNp8_SetSearchOrderBy

Sets the ORDERBY clause for the search results.

Syntax

```
bool FNp8_SetSearchOrderBy (StrParam)
```

Parameters

A string value of the ORDERBY clause that sorts search results. Smart parameters are supported. The clause is not validated until the FNP8_SearchAndDownload action is executed. The expected format of the parameter is - PROPERTY_NAME ASC|DESC

Returns

False if parameter is not a string. Otherwise, True.

Level

All levels.

Details

This action is *optional* before you call action FNP8_SearchAndDownload.

Set the ORDERBY clause for the search results. Set to empty string "" to reset a previously set value.

The parameter is not validated until the SearchAndDownload is executed.

Example 1

```
FNP8_SetSearchClass("Invoice")
FNP8_SetSearchDownloadDir("C:/mylocaldir")
FNP8_SetSearchOrderBy("DocumentTitle ASC")
FNP8_SearchAndDownload()
```

In this example, the results are sorted by DocumentTitle in ascending order.

Example 2

```
FNP8_SetSearchClass("Invoice")
FNP8_SetSearchDownloadDir("C:/mylocaldir")
FNP8_SetSearchOrderBy("DocumentTitle DESC")
FNP8_SearchAndDownload()
```

In this example, the results are sorted by DocumentTitle in descending order.

Parent topic: [FileNet P8 actions](#)

FNP8_SetSearchWhereClause

Sets the search criteria (WHERE clause).

Syntax

```
bool FNP8_SetSearchWhereClause (StrParam)
```

Parameters

A string value of the SQL search criteria (that is, the WHERE clause). The criteria should follow the standard that is used by FileNet P8 query, which generally conforms to the SQL-92 standard. Smart parameters are supported. The clause is not validated until the FNP8_SearchAndDownload action is executed.

Returns

False if parameter is not a string. Otherwise, True.

Level

All levels.

Details

This action is *optional* before you call action FNP8_SearchAndDownload.

Set the SQL search criteria (that is, the WHERE clause). Set to empty string "" to reset a previously set WHERE clause.

Note: Only parametric search is supported, not content-based (full-text) search.

The syntax of the WHERE clause is not validated until the FNP8_SearchAndDownload action is executed. The SQL search criteria must follow the IBM FileNet standard, which generally conforms to SQL-92, with extensions for IBM FileNet specific constructs.

Example

```
FNP8_SetSearchClass("Invoice")
FNP8_SetSearchDownloadDir("C:/mylocaldir")
FNP8_SetSearchWhereClause("DocumentTitle LIKE '2017-Invoices%'")
FNP8_SearchAndDownload()
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetDefineSecurityParentage

Specifies whether the scanned documents that are exported from Datacap to the FileNet Content Manager inherit security from the parent folder.

Syntax

```
bool FNP8_SetDefineSecurityParentage(StrParam)
```

Parameters

The FNP8_SetDefineSecurityParentage action takes a Boolean parameter: True Or False.

Returns

If the action is set to True, the security settings of the exported scanned documents are inherited from the parent folder. This is equivalent to the following API setting.

```
filenet.api.constant.DefineSecurityParentage=DEFINE_SECURITY_PARENTAGE
```

If the action is set to False, the security settings of the exported scanned documents are not inherited from the parent folder. This is equivalent to the following API setting.

```
filenet.api.constant.DefineSecurityParentage=DO_NOT_DEFINE_SECURITY_PARENTAGE
```

Level

All levels.

Details

By default, the `FNp8_SetDefineSecurityParentage` action is set to `False` and the security is not inherited from the parent folder.

Note: If the `FNp8_SetDefineSecurityParentage` action is not used, or if any invalid parameter is used, the default value `False` is used and security is not inherited from the parent folder.

Example

```
FNp8_SetDestinationFolder("@APPVAR(values/gen/P8Folder)")
FNp8_SetDefineSecurityParentage("True")
FNp8_Upload()
```

Parent topic: [FileNet P8 actions](#)

FNp8_SetDestinationFolder

Sets the destination folder for the documents to be uploaded.

Syntax

```
bool FNp8_SetDestinationFolder(StrParam)
```

Parameters

The path to the destination FileNet P8 folder in the selected Object Store where documents are to be uploaded. For example: `\1040EZ\`. The default value is the root folder of the specified target object. Smart parameters are supported.

Attention: The trailing backslash is optional. Subfolders of the root folder do not need to have a trailing backslash after the name. In some cases, a main folder of the object store requires a trailing backslash, if the object store is configured for it.

Returns

`False`, if `DC_P8_Server.dll` was not successfully installed on this computer, or either parameter value is missing or not a string. Otherwise, `True`.

Level

All levels.

Details

Sets the destination folder for the documents to be uploaded. See also `FNp8_CreateFolder`.

Note: This setting can be changed by a subsequent `FNp8_CreateFolder` action. If you call it with the folder name `Unfiled Documents` that is not case-sensitive, it uploads to the FileNet® special `Unfiled Documents` folder.

Example

```
FNp8_SetDestinationFolder("/1040EZ/")
```

Parent topic: [FileNet P8 actions](#)

FPN8_SetDocClassId

Sets the FileNet® P8 document class for the uploaded files.

Syntax

```
bool FNP8_SetDocClassId(StrParam)
```

»

Parameters

A string or Smart Parameter identifying the value of the Document Class ID.

The allowed predefined variables are: [@BATCHID](#), [@ID](#), [@STATUS](#), [@TYPE](#), [@VALUE](#), [@JOBID](#), [@JOBNAME](#), [@OPERATOR](#), [@STATION](#), [@TASKID](#), [@TASKNAME](#) and [@name](#).

«

Returns

False if DC_P8_Server.dll was not successfully installed on this computer, or if parameter is not a string. Otherwise, True.

Level

All levels.

Details

Sets the Document Class to be used in FileNet P8 for the documents being uploaded. If this action is not called, the default value of Document is used.

Example

```
FPN8_SetDocClassId("Document")
```

Parent topic: [FileNet P8 actions](#)

FPN8_SetDocTitle

Sets the document title for documents that you are uploading.

Syntax

```
bool FNP8_SetDocTitle(StrParam)
```

Parameters

String value of a Document Title or a predefined Smart Parameter variable. *Title* is an acceptable default parameter.

Smart parameters are supported.

Returns

False if DC_P8_Server.dll was not successfully installed on this computer, or if parameter is not a string. Otherwise, True.

Level

All levels.

Details

Sets the Document Title for documents being uploaded.

Example

```
FNp8_SetDocTitle("1040ez")  
or  
FNp8_SetDocTitle("@ID")
```

Parent topic: [FileNet P8 actions](#)

FNp8_SetFileMimeType

Associates a custom MIME type with a file extension for documents that are uploaded to the FileNet P8 system.

Member of namespace

FileNet P8

Syntax

```
bool FNp8_SetFileMimeType (StrParam)
```

Parameters

StrParam

The file extension and the associated MIME type separated by a comma.

Returns

Always True.

Level

All levels.

Details

Use this action to set the MIME (Multipurpose Internet Mail Extensions) type for uploaded documents with the specified file extension. This action overrides the library's default mapping of the file extension to a MIME type.

The following table shows the default mappings of file extension to MIME type. Any unlisted file type is mapped to the generic MIME type *application/octet-stream*.

ai	application/illustrator
avi	video/x-msvideo
bmp	image/bmp
doc	application/msword
docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document
dwg	image/vnd.dwg
dxf	image/vnd.dxf
gif	image/gif
htm	text/html
html	text/html
ico	image/x-icon
jpe	image/jpeg
jpeg	image/jpeg
jpg	image/jpeg
jpgv	video/jpeg
m4v	video/x-m4v
mov	video/quicktime
mp4	video/mp4
mpeg	video/mpeg
mpg	video/mpeg
mpp	application/vnd.ms-project
mpv	video/mpv
msg	application/msoutlook
ogg	application/ogg
pcx	image/x-pcx
pdf	application/pdf
pic	image/x-pict
png	image/png
ppt	application/vnd.ms-powerpoint
pptx	application/vnd.openxmlformats-officedocument.presentationml.presentation
qt	video/quicktime
rgb	image/x-rgb
rtf	text/richtext
tif	image/tiff

tiff	image/tiff
txt	text/plain
vsd	application/vnd.visio
wmd	application/x-ms-wmd
wmv	video/x-ms-wmv
wmx	video/x-ms-wmx
xls	application/vnd.ms-excel
xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
xml	text/xml

Example:

```
FNP8_SetFileMimeType(".rar, application/x-rar-compressed")
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetFileType

Assigns the file type for the files that are uploaded.

Syntax

```
bool FNP8_SetFileType (StrParam)
```

Parameters

StrParam

A string that identifies the file type. The valid values are as follows:

- doc
- docx
- gif
- jpe
- jpeg
- jpg
- msg
- pdf
- ppt
- tif
- xls
- xlsx
- zip

The default file type is TIF.

Returns

Always returns True.

Level

Batch or Document level.

Details

Use this action to identify the file type of the files that will be uploaded to FileNet® P8.

Example

```
 FNP8_SetFileType ("jpg")
 FNP8_Upload ()
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetKeyProperty

Sets the update key to a FileNet® document property and its corresponding property value.

Syntax

```
bool FNP8_SetKeyProperty (StrParam)
```

Parameters

Strings values identify a FileNet document class property and its corresponding property value. The document class property name and its value must exist on the destination FileNet P8 Object Store. The property value must also be unique because the call to the FNP8_UpdateProperties action returns only one value. Smart parameters are allowed for the second parameter.

Returns

False if either parameter is blank or if the value parameter is invalid. Otherwise, True.

Level

Batch, Document or Page level.

Details

This action sets a key that is used by the UpdateProperties action. The key is a FileNet document property id and its corresponding value. The UpdateProperties action uses the key to search for an existing FileNet document. If the document is found, the properties on the document are updated with the values specified in the SetProperty actions.

Example

```
 FNP8_SetKeyProperty ("DCKey,@DCKey")
 FNP8_SetProperty ("FNProperty,@SomeValue")
 FNP8_SetKeyProperty ("@FNProperty2,@SomeValue2")
 FNP8_UpdateProperties ()
```

This example uses the parameters of FNP8_SetKeyProperty("DCKey,@DCKey") to search for a FileNet document on an object store. If the FileNet document is found, it is assigned the values that are

specified in the SetProperty action when the UpdateProperties action is called. The UpdateProperties action performs actions only on the first document that is returned. If more than one document matches the criteria that is specified in the SetKeyProperty action, only the first document is updated.

Parent topic: [FileNet P8 actions](#)

FNP8_SetLocale

Identifies the locale on the target FileNet® P8 system.

Syntax

```
bool FNP8_SetLocale(StrParam)
```

Parameters

Locale value accepted by the FileNet P8 Web Service. The default value is en_US.

Locales are represented by 2-letter ISO 639 Language Codes and 2-letter ISO 3166 Country Codes separated by a _ character. For example: en_US or de_DE.

Please consult these ISO documents to determine your locale value.

Returns

False, if DC_P8_Server.dll was not successfully installed on this computer, or either parameter value is missing or not a string. Otherwise, True.

Level

All, but generally at the Batch level.

Details

Sets the Locale (the language and language conventions such as date format) used on the P8 server. This action is only required if the FileNet P8 repository uses a locale other than US English.

Example

```
FNP8_SetLocale("en_US")
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetMultiValueProperty

Sets the values in a multi-value property.

Syntax

```
bool FNP8_SetMultiValueProperty(StrParam)
```

Parameters

A comma separated string consisting of these three values:

1. Property ID
2. Property Value. Smart parameters are supported for this parameter only.
3. An optional Property Type. The default is a String. Refer to the FileNet® P8 documentation for a list of Property Types.

Returns

False if the ID or Value parameters are missing or if the specified property is not a multi-value property. Otherwise True.

Level

All levels.

Details

This action sets the property of a FileNet P8 multi-value property. It can be called multiple times.

Example

```
FNP8_SetDocClassId("MyFilenetClass")
FNP8_SetDocTitle("MyFilenetClass Documents")
FNP8_SetProperty("CustomerName,@D.CustomerName")
FNP8_SetMultiValueProperty("InvoiceList,@D.InvoiceList")
FNP8_SetProperty("ScanStation,@STATION")
FNP8_SetProperty("ScanOperator,@OPERATOR")
FNP8_SetFileType("pdf")
FNP8_Upload()
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetProperty

Sets the designated FileNet® P8 property to a specified value.

Syntax

```
bool FNP8_SetProperty(StrParam)
```

Parameters

The following comma-separated string values:

1. Property ID: the name of an existing document property in the FileNet library (equivalent to a document index field).
2. Property Value: the value to assign to the associated Property ID.
3. Optional property type. If this parameter is not specified, the property type will default to a 'string'. Supported types are: Binary, Boolean, DateTime, Float, ID, Integer, Object and String.

Smart Parameters are supported for values.

Returns

False if either parameter is blank or if the value parameter is invalid. Otherwise True.

Attention: If the action returns False, the action directs the Rulerunner task to finish with a status of Abort. The task will also abort if more than one value is assigned to the Smart Parameter variable. If a ruleset calls the FNP8_SetProperty action more than once, using the same Property ID as the opening parameter, FileNet P8 assumes that the second parameter is multi-value and assigns that value to the property.

Level

All levels.

Details

Sets the designated FileNet property to a specified value. This is equivalent to setting an index value for a document in other document management systems.

Example

```
FNP8_SetProperty("DocumentTitle, @ID")
FNP8_Upload()
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetPropertyEx

Sets the designated FileNet® P8 property to a specified value.

Syntax

```
bool FNP8_SetPropertyEx(StrParam)
```

Parameters

The following comma-separated string values:

1. Property ID: the name of an existing document property in the FileNet P8 library (equivalent to a document index field).
2. Property Value: the value to assign to the associated Property ID.
3. Optional property type. If this parameter is not specified, the property type defaults to a string. Supported types are: Binary, Boolean, DateTime, Float, ID, Integer, Object and String.

Smart Parameters are supported for values.

Returns

False if either parameter is blank or if the value of the parameter is invalid. Otherwise True.

Attention: If the action returns False, the action directs the Rulerunner task to finish with a status of Abort. The task will also abort if more than one value is assigned to the Smart Parameter variable.

Important: Any ruleset that calls the FNP8_SetPropertyEx action more than once, using the same Property ID as the opening parameter, the property value will not create a multivalued property. Unlike the behavior of the FNP8_SetProperty action, it replaces the existing property value that was set up by any previous ruleset.

Level

All levels.

Details

Sets the designated FileNet property to a specified value. This is equivalent to setting an index value for a document in other document management systems.

Example

```
FNp8_SetPropertyEx("DocumentTitle, @ID")
FNp8_Upload()
```

Parent topic: [FileNet P8 actions](#)

FNp8_SetRetry

Sets the number of automatic upload retries.

Syntax

```
bool FNp8_SetRetry(StrParam)
```

Parameters

A integer value identifies the amount of retries performed if the upload failed. Smart parameters are supported.

Returns

False if non-numeric parameter specified. Otherwise True.

Level

Batch, Document or Page level.

Details

Use this action to set the amount of retries for the FileNet® P8 upload actions. If the upload to FileNet fails, the upload action will immediately be retried the number of times specified by the FNp8_SetRetry action.

If this action is not called prior to a FileNet P8 upload action, the default amount of retries is 0.

Example

```
FNp8_SetRetry("3")
NP8_Upload()
```

FNp8_SetRetry("3") causes the FNp8_Upload to initiate 3 upload attempts if the upload to FileNet fails.

Parent topic: [FileNet P8 actions](#)

FNp8_SetTargetClassID

Sets the FileNet® P8 document class for uploaded documents.

Syntax

```
bool FNP8_SetTargetClassID(StrParam)
```

Parameters

Specifies the repository type. The valid values are ObjectStore and FileStore.

Returns

False, if DC_P8_Server.dll was not successfully installed on this computer, or either parameter value is missing or not a string. Otherwise, True.

Level

All levels.

Details

Sets the top-level repository type. If this action is not called, the default value of ObjectStore is used. FileStore is an alternative repository type.

Example

```
FNP8_SetTargetClassID("ObjectStore")
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetTargetObjectID

Sets the name of the Object Store in which documents are stored.

Syntax

```
bool FNP8_SetTargetObjectID (StrParam)
```

Parameters

String value of the Object ID. Smart parameters are supported.

Returns

False, if DC_P8_Server.dll was not successfully installed on this computer, or either parameter value is missing or not a string. Otherwise, True.

Level

All, but generally at the Batch level.

Details

Provides the name of the Object Store that you wish to store your documents in.

Example

```
FNP8_SetTargetObjectID("AP_ObjectStore")
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetTimeout

Sets the timeout for the FileNet® P8 web service in milliseconds.

Syntax

```
bool FNP8_SetTimeout (StrParam)
```

Parameters

A single parameter identifying the timeout in milliseconds for the FileNet P8 Web Service. The default timeout is 600000 milliseconds (600 seconds).

Returns

False if DC_P8_Server.dll was not successfully installed on this computer, or if the parameter is not formatted as a valid integer value. Otherwise, True.

Level

All, but generally at the Batch level.

Details

This action sets the timeout in milliseconds for the FileNet P8 Web Service. This action should be called before the FNP8_Upload action.

Example

```
FNP8_SetTimeout (90000)
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetUploadMode

Sets the upload mode.

Syntax

```
bool FNP8_SetUploadMode (StrParam)
```

Parameters

A string or Smart Parameter identifying the page level variable where file name is stored. If this action is not called the value defaults to blank and regular upload logic applied. Smart parameters are supported.

For example FNP8_SetUploadMode("ParentImage") uploads a file with the name stored in *ParentImage* variable on the page level.

Returns

Always True.

Level

Batch, Document or Page level.

Details

Use this action to identify the files that will be uploaded to FileNet® P8.

Example

```
 FNP8_SetUploadMode ("ParentImage")
 FNP8_Upload ()
```

Parent topic: [FileNet P8 actions](#)

FNP8_SetURL

Sets the URL for the FileNet® P8 Server that is used.

Syntax

```
bool FNP8_SetURL (StrParam)
```

Parameters

A single parameter that identifies the URL for the IBM® FileNet P8 Server. The IP port is typically :9080 (IBM WebSphere®) and the URL typically ends in /WSDL.

Returns

False, if DC_P8_Server.dll was not successfully installed on this computer, or if the parameter is not formatted as a valid URL. Otherwise, True.

Attention: This action returns True whether the URL is to an operating IBM FileNet P8 Server or not. If not, an error is reported when the Upload or UploadDir action runs. You can check that the URL is correct and operating by displaying it in any web browser.

Level

All, but generally at the Batch level.

Details

This action sets the URL for the IBM FileNet P8 IBM FileNet P8 Server. This action must be called before any other FileNet P8 upload actions.

Example

```
 FNP8_SetURL ("http://myp8server:9080/wsi/FNCEWS40MTOM/")
```

Important: The address can be case-sensitive and the trailing forward-slash "/" must be present. To confirm that the address reaches a working IBM FileNet P8 Server, browse to the address after you add

wsdl to the end of the URL address, as in the following example,
<http://myp8server:9080/wsi/FNCEWS40MTOM/wsdl>

Parent topic: [FileNet P8 actions](#)

FNP8_UpdateProperties

Updates the properties of an existing FileNet® P8 document by using the data that is passed into the FNP8_SetProperty action.

Syntax

```
bool FNP8_UpdateProperties()
```

Parameters

Uses the key specified in the FNP8_SetKeyProperty action to search for an existing document within a FileNet P8 Case Manager Object Store and updates the P8 document's properties as specified by the FNP8_SetProperty action.

Returns

True if the update is successful, False if the action is unable to update the document.

Level

Batch, Document or Page level.

Details

Use this action to update an existing FileNet P8 document's properties. The UpdateProperties action should only be used to update properties for an existing document. If a new document is created use the Upload action.

Example

```
FNP8_SetKeyProperty ("DCKey,@DCKey)  
FNP8_SetProperty("AGIncome, @AGIncome")  
FNP8_SetProperty("TaxYear,@TaxYear)  
FNP8_UpdateProperties()
```

This example will first set the key document property name and value to search for a document, the next two actions indicate which properties and values that the FileNet P8 document should be updated with. The FNP8_UpdateProperties will invoke the actual update. If more than one document matches the criteria specified in the SetKeyProperty action, only the first document will be updated.

Parent topic: [FileNet P8 actions](#)

FNP8_Upload

Uploads the batch images to the FileNet® P8 repository.

Syntax

```
bool FNP8_Upload()
```

Parameters

None.

Returns

False if the upload is not successful, or the action was applied to the Field level. Otherwise, True. If successful, each page uploaded will have a variable *Doc_ID* set to the FileNet document identifier.

Attention: The action directs the Rulerunner task to finish with a status of Aborted.

Level

Batch, Document or Page level.

Details

Uploads image files from the Datacap batch to the specified destination folder in the FileNet library.

When called at the Batch or Document level, attempts to upload a multipage TIF image file named ObjID.tif, where ObjID is the DCO Object ID of the Batch or Document. If such a file does not exist, tries to upload all page images in the Batch or Document.

When called at the Page level, uploads the image file for that page. It is important that any required properties are set before calling FNP8_Upload such as FNP8_SetDocTitle otherwise an upload error could occur.

Example

```
FNP8_SetURL("http://server:port//wsi/FNCEWS40MTOM/")
FNP8_Login("user,password")
FNP8_SetLocale("en-us")
FNP8_SetDocClassId("MyFileNetDocClass")
FNP8_SetDocTitle("Document Title")
FNP8_SetProperty("CustomerName,@D.CustomerName")
FNP8_SetMultiValueProperty("InvoiceList,@D.InvoiceList")
FNP8_SetProperty("ScanStation,@STATION")
FNP8_SetProperty("ScanOperator,@OPERATOR")
FNP8_SetFileType("pdf")
FNP8_Upload()
```

Parent topic: [FileNet P8 actions](#)

FNP8_UploadDir

Uploads all of the images in the folder to the specified destination folder.

Syntax

```
bool FNP8_UploadDir(StrParam)
```

Parameters

Two comma-separated String values:

1. The full path of the source folder that contains the images to be uploaded. For example: C:\images.

2. A Boolean value to indicate whether images must be deleted from the source folder. False: images will not be deleted from the source folder after they are uploaded. True: images will be deleted from the source folder after they are uploaded.

Returns

False, if the upload is not successful. Otherwise, True.

Attention: If the action returns False, the action directs the Rulerunner task to finish with a status of `Aborted`.

Level

Batch or Document level.

Details

Uploads all images in the folder you enter as the first parameter to the specified destination folder.

This action is an alternative to `FNP8_Upload` if you want to upload images that are not within a Datacap batch.

Example

```
FNP8_UploadDir("C:\images,False")
```

This example leaves the images in the source folder after they are uploaded.

```
FNP8_UploadDir("C:\images,True")
```

This example deletes the images in the source folder after they are uploaded.

Parent topic: [FileNet P8 actions](#)

FingerprintMaintenance actions

Use the `FingerprintMaintenance` actions to delete fingerprints from the fingerprint library of the application.

The `FingerprintMaintenance` actions open and close connections to the fingerprint database, specify the folder that contains the fingerprints to be deleted, and deletes those fingerprints.

- [CloseDatabase](#)
Closes connection to the fingerprint database and saves the Setup DCO.
- [DeleteFingerprint](#)
Deletes specified fingerprint.
- [DeleteFingerprints](#)
Deletes all fingerprints that are returned by the SQL statement in the parameter
- [OpenDatabase](#)
Opens a connection to the fingerprint database.
- [SetFingerprintFolder](#)
Specifies the folder containing the fingerprint files.

Parent topic: [Global actions](#)

CloseDatabase

Closes connection to the fingerprint database and saves the Setup DCO.

Member of namespace

FingerprintMaintenance

Syntax

```
bool CloseDatabase ()
```

Parameters

None.

Returns

Always True.

Level

Any level.

Details

Closes the open database connection and saves the Setup DCO if position information was deleted. The Setup DCO is not modified if FPXML files are used, or if no fingerprints were deleted.

Example:

```
CloseDatabase ()
```

Parent topic: [FingerprintMaintenance actions](#)

DeleteFingerprint

Deletes specified fingerprint.

Member of namespace

FingerprintMaintenance

Syntax

```
bool DeleteFingerprint (string ID)
```

Parameters

ID

Type: string

ID of fingerprint to delete

Parameters

The ID of the fingerprint to delete.

Returns

Always True.

Level

Any level.

Details

Deletes the CCO, TIFF and FPXML files for the specified fingerprint. Also deletes the database record and position information from the DCO. A smart parameter may be used.

Example:

```
DeleteFingerprint ("1002")
```

Parent topic: [FingerprintMaintenance actions](#)

DeleteFingerprints

Deletes all fingerprints that are returned by the SQL statement in the parameter

Member of namespace

FingerprintMaintenance

Syntax

```
bool DeleteFingerprints (string SQL)
```

Parameters

SQL

Type: string

SQL statement to return a recordset of fingerprints to be deleted.

Parameters

SQL statement that is used to return a recordset of fingerprints that must be deleted.

Returns

Always True.

Level

Any level.

Details

Deletes all fingerprints that are returned by the query that is specified in the parameter. A smart parameter can be used.

Example:

```
Access - DeleteFingerprints("SELECT * FROM Template WHERE tp_LastHit <
dateadd("d",-30,NOW) ")
SQL - DeleteFingerprints("SELECT * FROM Template WHERE tp_LastHit <
dateadd(d,-2,GETDATE()) ")
Oracle- DeleteFingerprints("SELECT * FROM Template WHERE tp_LastHit <
TRUNC(SYSDATE) - 30")
```

Parent topic: [FingerprintMaintenance actions](#)

OpenDatabase

Opens a connection to the fingerprint database.

Member of namespace

FingerprintMaintenance

Syntax

```
bool OpenDatabase (string ConnectionString)
```

Parameters

ConnectionString
Type: string
Fingerprint database connection string.

Parameters

Fingerprint database connection string.

Returns

True, if the connection is established. False, if the connection is not established.

Level

Any level.

Details

Opens a connection to the fingerprint database by using the connection string provided in the parameter. A smart parameter can be used.

Example:

```
OpenDatabase ("provider=microsoft.jet.oledb.4.0;
data source=c:\datacap\apt\APTFingerprint.mdb;
persist security info=false")
```

This example obtains the connection string set in the Application Manager.

```
OpenDatabase("@APPVAR(*/*dco_*[1]/fingerprintconn:cs)")
```

Parent topic: [FingerprintMaintenance actions](#)

SetFingerprintFolder

Specifies the folder containing the fingerprint files.

Member of namespace

FingerprintMaintenance

Syntax

```
bool SetFingerprintFolder (string Folder)
```

Parameters

Folder

Type: string

Folder containing the fingerprints.

Parameters

Folder path where the fingerprints are located.

Returns

Always True.

Level

Any level.

Details

Specifies the folder containing the fingerprint files. A smart parameter may be used.

Example:

```
SetFingerprintFolder("C:\Datacap\APT\fingerprint")
```

This example obtains the path set in the Application Manager.

```
SetFingerprintFolder("@APPPATH(*/*fingerprint)")
```

Parent topic: [FingerprintMaintenance actions](#)

FPXML actions

Use the FPXML actions to store zone coordinates in an external XML file instead of the document hierarchy (setup DCO). These actions are useful for porting fingerprints between systems or to avoid making frequent

modifications to the document hierarchy.

The FPXML actions can load zone information for a fingerprint, set the type of Details and LineItem fields, and set the location for the fingerprint XML files.

- [ReadZonesFPX](#)
Loads position information for current fingerprint
- [SetDetailsAndLineitemPairFPX](#)
Sets the type of the special Details and Lineitem fields.
- [SetDirectoryFPX](#)
Sets the directory in which fingerprint xml files should be read and written to.
- [WriteZoneFPX](#)
Writes the positions of a field to Fingerprint XML
- [WriteZonesFPX](#)
Writes position information for all fields of the page

Parent topic: [Global actions](#)

ReadZonesFPX

Loads position information for current fingerprint

Member of namespace

FPXML

Syntax

```
bool ReadZonesFPX ()
```

Parameters

None.

Returns

False, if the fingerprint.xml cannot be loaded or if SetDirectoryFPX has not been called to set the location of the fingerprint XML files. Otherwise, True.

Level

Page or Field level.

Details

This action reads the field zone information from the fingerprint XML file associated with the fingerprint for the current page object. This action is similar to the ReadZones action in the Zones library. The primary difference is that ReadZones reads the zone information stored in the setup DCO and ReadZonesFPX reads the zone information from the associated fingerprint XML file. Typically this action is called after the CreateFields action but before field recognition. ReadZonesFPX loads position information for each node in the calling object and it's children. Pre-adjusts these values based on offset information stored in the *Image_Offset* variable at any node level.

The offset value is applied to all child objects of a node where an *Image_Offset* variable is found; unless overwritten by a child node also having an *Image_Offset* value to apply.

Position information is based on the Fingerprint XML position for the parent page's fingerprint ID.

If Enable FPXML setting is checked in the application settings Main tab in the Datacap Application Manager, Datacap Studio places field zone information into a fingerprint XML file instead of placing the zone information into the applications setup DCO. One FPXML file is created per fingerprint. Depending on application needs, using fingerprint XML files might have advantages. Refer to Datacap documentation for more information about using fingerprint XML files.

If the application is using FPXML files, then use this action to load the zones.

Example:

```
SetDirectoryFPX("@APPPATH(* /fingerprint) ")
ReadZonesFPX()
```

Parent topic: [FPXML actions](#)

SetDetailsAndLineitemPairFPX

Sets the type of the special Details and Lineitem fields.

Member of namespace

FPXML

Syntax

```
bool SetDetailsAndLineitemPairFPX (StrParam)
```

Parameters

Two comma separated parameters:

- Detail field type: type of the field that is the parent to the line item field, often called Details.
- LineItem field type: type of the line item field which is the parent of the fields to be captured, often called Lineitem.

Smart parameters are supported.

Returns

False if one of the parameters is missing. Otherwise True.

Level

Any level.

Details

Use this action if your page contains Details and Lineitems fields, as is common with an invoice page. The action sets the type of the special Details and Lineitem fields. These fields, unlike other field types, are handled

by the ReadZonesFPX and WriteZonesFPX actions in a special way that is compatible with the Locate actions framework.

If your page has Details and Lineitems, then this action is required to identify the types so the position information is properly saved. Using SetDetailsAndLineitemPairFPX prior to calling WriteZonesFPX causes WriteZonesFPX to save line item field positions to the fingerprint file. The second parameter is the type of the line item field which is the parent of the fields to be captured and is often called Lineitem.

If your page contains Detail and Lineitem fields, this action also must be used prior to calling ReadZonesFPX.

Details are the parent object of the line item fields. Details are a collection of lineitems and each line item has a collection of fields. For example, in a typical invoice application a single lineitem can contain the fields: quantity, item number, description, unit price, and total.

Example:

```
SetDetailsAndLineitemPairFPX("Details,Lineitem")
```

```
ReadZonesFPX()
```

In this example, the details field is called "Details" and the line items are called "Lineitem".

An example DCO hierarchy may look something like this:

```
- Details
--- Lineitem
----- Qty
----- ItemID
----- ItemDesc
----- Price
----- LineTotal
```

Parent topic: [FPXML actions](#)

SetDirectoryFPX

Sets the directory in which fingerprint xml files should be read and written to.

Member of namespace

FPXML

Syntax

```
bool SetDirectoryFPX (StrParam)
```

Parameters

A string identifying the Fingerprint directory. This is the directory that contains the FPXML files.

Smart parameters are supported.

Returns

False if the parameter is missing. Otherwise True.

Level

Any level.

Details

This action sets the directory from which fingerprint XML files will be written or read.

This action must be called to identify the FPXML directory before calling ReadZonesFPX or WriteZonesFPX.

Example:

```
SetDirectoryFPX("@APPPATH(* /fingerprint)")  
ReadZonesFPX()
```

This example uses the @APPPATH smart parameter to obtain the directory from the application service.

```
SetDirectoryFPX("C:\Datacap\APT\fingerprint")  
ReadZonesFPX()
```

This example hard codes the directory, which is fine but could make the application less portable.

Parent topic: [FPXML actions](#)

WriteZoneFPX

Writes the positions of a field to Fingerprint XML

Member of namespace

FPXML

Syntax

```
bool WriteZoneFPX (StrParam)
```

Parameters

Field name (required only when the action is called at the page level)

Returns

False if there is there is not a fingerprint directory set, or the fingerprint ID cannot be determined, if the field in the parameter cannot be found, if called at the page level without a parameter, or if there is a problem opening or saving the fingerprint XML file. Otherwise True.

Level

Page and Field

Details

When called at the field level, writes position information for each node in the calling object and its children.

When called at the page level, writes position information the specified field and its sub fields.

Parent topic: [FPXML actions](#)

WriteZonesFPX

Writes position information for all fields of the page

Member of namespace

FPXML

Syntax

```
bool WriteZonesFPX (StrParam)
```

Parameters

Three comma separated parameters:

- Fingerprint host name: This is the fingerprint class name and can be seen in the Zones tab of Datacap Studio. It is used to group or classify fingerprints. If this value is not provided, then a blank class name will be used.
- Fingerprint host id: This is an internal numeric value that is associated with the fingerprint class name. If a value is not provided, a blank value is used.
- Fingerprint page type: This is the page type that corresponds to the fingerprint for the page. If this value is not provided, it uses the page type of the current page.

Smart parameters are supported. All parameters are optional. If you provide only the second or third parameter, you must supply an empty value for the previous parameters.

Returns

False, if there is a problem saving to fingerprint XML file or if not called at the page level, if the page does not have a fingerprint assigned, or a fingerprint directory has not been set. Otherwise True.

Level

Page level only.

Details

This action writes out all of the field positions for the current page to the fingerprint XML file. Typically this action is used for applications that are using fingerprint XML files and new pages are dynamically zoned at runtime. After the zoning of the page has been completed, then this action is used to write out the zone information to the FPXML file, allowing subsequent pages of this type to be processed using the saved zone information.

If the fingerprint directory that had been previously specified using SetDirectoryFPX does not exist, it is created, providing that the parent directory already exists.

Fields with empty positions are not written to the fingerprint XML file.

Example:

```
SetDirectoryFPX("@APPPATH(* /fingerprint) ")
WriteZonesFPX("MyFPClass,,MyPageType")
```

This example writes out the zones for the current page and sets the fingerprint class to "MyFPClass" and the page type to "MyPageType".

```
SetDirectoryFPX("@APPPATH(* /fingerprint) ")
WriteZonesFPX("MyFPClass,,@P.MyType")
```

This example writes out the zones for the current page and sets the fingerprint class to "MyFPClass" and the page type to the value contained by the page variable "MyType".

Parent topic: [FPXML actions](#)

Grayscale actions

Use the Grayscale action to convert grayscale TIFF images to black-and-white.

The ConvertGraytoBW action can convert color dropout forms, including medical claims, scanned in grayscale to black and white. The action categorizes each pixel in the image as either foreground (black), dropout (red / light gray), or background (white), to produce a final black and white image without the dropout color.

- [ConvertGraytoBW](#)
Converts Grayscale TIFF files to black and white TIFF files.

Parent topic: [Global actions](#)

ConvertGraytoBW

Converts Grayscale TIFF files to black and white TIFF files.

Syntax

```
bool ConvertGraytoBW ()
```

Parameters

None.

Returns

False, if a rule set with this action is bound to a Field object of the Document Hierarchy. Otherwise, True.

If the input image is not grayscale, or if there is an error in processing, the image is not converted and the action returns False.

Level

Batch, Document or Page level. If called at the batch level, all images will be converted.

If called at the document level all of the pages within the document will be converted.

If called at the page level, the single page will be converted.

Details

This action converts grayscale TIFF files into Black and White TIFF files.

The ConvertGraytoBW action is especially good at converting color dropout forms, including medical claims, scanned in grayscale to black and white. The action categorizes each pixel in the image as either foreground (black), dropout (red / light gray), or background (white), to produce a final black and white image without the dropout color.

Attention: The action renames the original grayscale image using the same base file name, but replaces the .tif filename extension with .tis.

Example

```
ConvertGraytoBW ()
```

Parent topic: [Grayscale actions](#)

HandwritingRecognition actions

Use the Handwriting Recognition actions to perform cursive recognition of the configured fields using the Parascript FormXtra Recognition Engine.

The Handwriting Recognition actions are described in the following table.

- [Recognize](#)
Recognizes the content within the configured fields.
- [SetAddressApartmentZone](#)
Sets the name of the field that contains the zone for the apartment part of the address.
- [SetAddressCityStateZipZone](#)
Sets the name of the field that contains the zone for the City, State, and Zip part of the address.
- [SetAddressCityStateZone](#)
Sets the name of the field that contains the zone for the City and State part of the address.
- [SetAddressCityZone](#)
Sets the name of the field that contains the zone for the City part of the address.
- [SetAddressStateZone](#)
Sets the name of the field that contains the zone for the State part of the address.
- [SetAddressZipZone](#)
Sets the name of the field that contains the zone for the Zip code (postal code) part of the address.
- [SetAmountFormat](#)
Sets the format of the amount fields.
- [SetBoxRemovalMode](#)
Sets the type of box removal to be performed on the field.
- [SetCreditCardFormat](#)
Sets the format of credit card number fields.
- [SetDateFormat](#)
Sets the date format for date fields.
- [SetDeskew](#)
Turns on or off the deskew on the field snippet prior to recognition.
- [SetFieldType](#)
Sets the type of the field to read.
- [SetFullNameFormat](#)
Sets the format of full name fields.
- [SetLanguage](#)
Sets the recognition language for the fields.
- [SetLineRemovalMode](#)
Sets the type of line removal to be performed on the field.

- [SetMultiLineMode](#)
Sets which lines to extract from multi-line fields.
- [SetNoiseRemoval](#)
Turns on or off the noise removal on the field snippet prior to recognition.
- [SetPostalDatabase](#)
Sets the path of the postal database for address recognition.
- [SetProcessingMode](#)
Sets whether to favor speed or accuracy when performing recognition.
- [SetSpecialCharacterSet](#)
Sets additional characters to add to the default field character set.
- [SetTemplate](#)
Sets the format of the field text to read.
- [SetValidLength](#)
Sets the length range for the field.
- [SetValidValues](#)
Sets the length range for the field.
- [SetVocabulary](#)
Sets the vocabulary file to use for recognizing fields.
- [SetWritingStyle](#)
Sets the writing style of the field.

Parent topic: [Global actions](#)

Recognize

Recognizes the content within the configured fields.

Member of namespace

HandwritingRecognition

Syntax

```
bool Recognize ()
```

Returns

True always.

Level

Page.

Details

Performs recognition of the configured fields. After you have configured all the fields to be recognized, you must call this action to produce recognition results.

Example:

```
Field Level ruleset
```

```
SetFieldType  
SetWritingStyle
```

Page Level Ruleset

Recognize ()

The page level ruleset in this example is attached to the page's "Close" to ensure that all fields are configured before calling Recognize. Alternatively, the Recognize action could also be called in a separate ruleset, after the page fields have been configured.

Parent topic: [HandwritingRecognition actions](#)

SetAddressApartmentZone

Sets the name of the field that contains the zone for the apartment part of the address.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetAddressApartmentZone (string param)
```

Parameters

The string value representing name of the field that contains the zone for the apartment part of the address. Smart parameters are supported.

Returns

True always.

Level

Field.

Details

Sets the name of the field that contains the zone for the apartment part of the address. This is required when recognizing fields of type "Address" and multiple zones are used to recognize the address.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType (8)  
SetAddressApartmentZone ("ApartmentField")
```

The example above sets the field type to address. The field containing the apartment zone, ApartmentField in this example, is set using the SetAddressApartmentZone action. Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset)

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetAddressCityStateZipZone

Sets the name of the field that contains the zone for the City, State, and Zip part of the address.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetAddressCityStateZipZone (string param)
```

Parameters

The string value representing name of the field that contains the zone for the City, State, and Zip part of the US address. Smart parameters are supported.

Returns

True always.

Level

Field.

Details

Sets the name of the field that contains the zone for the City, State, and Zip part of the address. This is required when recognizing fields of type "Address" and multiple zones are used to recognize the address.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType(8)  
SetAddressCityStateZipZone("CityStateZipField")
```

The example above sets the field type to address. Assuming the field zoned for the City, State, and Zip zone is called CityStateZipField, this field is then specified using the SetAddressCityStateZipZone action. When recognition runs, the field CityStateZipField is recognized and its text value is set with the City, State and Zip value. The original field configured as the address field has DCO variables created that contain the City, State, Zip and ZipFull.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetAddressCityStateZone

Sets the name of the field that contains the zone for the City and State part of the address.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetAddressCityStateZone (string param)
```

Parameters

The string value representing name of the field that contains the zone for the City and State part of the address. Smart parameters are supported.

Returns

True always.

Level

Field.

Details

Sets the name of the field that contains the zone for the City and State part of the address. This is required when recognizing fields of type "Address" and multiple zones are used to recognize the address.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType(8)
SetAddressCityStateZone("CityStateField")
SetAddressZipZone("ZipCodeField")
```

The example above sets the field type to address. The field containing the City, State zone is configured. Finally the field containing the Zip zones is set.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetAddressCityZone

Sets the name of the field that contains the zone for the City part of the address.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetAddressCityStateZone (string param)
```

Parameters

The string value representing name of the field that contains the zone for the City part of the address. Smart parameters are supported.

Returns

True always.

Level

Field.

Details

Sets the name of the field that contains the zone for the City part of the address. This is required when recognizing fields of type "Address" and multiple zones are used to recognize the address.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType(8)
SetAddressCityZone("CityField")
SetAddressStateZone("StateField")
SetAddressZipZone("ZipCodeField")
```

The example above sets the field type to address. The field containing the City, State, and Zip zones are identified using the SetAddress actions.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

Recognize()

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetAddressStateZone

Sets the name of the field that contains the zone for the State part of the address.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetAddressStateZone (string param)
```

Parameters

The string value representing name of the field that contains the zone for the State part of the address. Smart parameters are supported.

Returns

True always.

Level

Field.

Details

Sets the name of the field that contains the zone for the State part of the address. This is required when recognizing fields of type "Address" and multiple zones are used to recognize the address.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType(8)
SetAddressCityZone("CityField")
SetAddressStateZone("StateField")
SetAddressZipZone("ZipCodeField")
```

The example above sets the field type to address. The field containing the City, State, and Zip zones are identified using the SetAddress actions.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetAddressZipZone

Sets the name of the field that contains the zone for the Zip code (postal code) part of the address.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetAddressZipZone (string param)
```

Parameters

The string value representing name of the field that contains the zone for the Zip code part of the address. Smart parameters are supported.

Returns

True always.

Level

Field.

Details

Sets the name of the field that contains the zone for the Zip code (postal code) part of the address. This is required when recognizing fields of type "Address" and multiple zones are used to recognize the address.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType (8)
SetAddressCityZone ("CityField")
SetAddressStateZone ("StateField")
SetAddressZipZone ("ZipCodeField")
```

The example above sets the field type to address. The field containing the City, State, and Zip zones are identified using the SetAddress actions.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetAmountFormat

Sets the format of the amount fields.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetAmountFormat (string param)
```

Parameters

The integer value representing the amount format. The following values are supported:

- 0 - Positive (default)
- 1 - Positive or negative

Returns

True always.

Level

Field.

Details

Sets the format of the amount fields.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType (32)  
SetFullNameFormat (1)
```

The example above sets the field type to amount, and sets the amount format to positive or negative amount.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetBoxRemovalMode

Sets the type of box removal to be performed on the field.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetBoxRemovalMode (string param)
```

Parameters

The integer value representing the type of box removal. The following are the valid values:

- 0 - None (default)
- 1 - Remove box
- 2 - Remove box and around
- 3 - Remove combs

Returns

True always.

Level

Field.

Details

Sets the type of box removal to be performed on the field.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetBoxRemovalMode (1)
SetFieldType (1)
SetMultiLineMode (2)
```

The example above sets line removal to remove boxes, then sets the field type to alpha, and finally sets the recognition to read only the second line.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetCreditCardFormat

Sets the format of credit card number fields.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetCreditCardFormat (string param)
```

Parameters

The integer value representing the format of credit card number fields. The following values are supported:

- 0 - Number (default)
- 1 - Number and expiration date

Returns

True always.

Level

Field.

Details

Sets the format of credit card number fields.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType (256)
SetCreditCardFormat (1)
```

The example above sets the field type to credit card number, and sets the format to read number and expiration date.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetDateFormat

Sets the date format for date fields.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetDateFormat (string param)
```

Parameters

The integer value representing the date format. The following values are supported:

- 0 - MMDDYYYY. The numeric month/day/year format. For example, 12/31/2010 or 12/31/10.
- 1 - MMMMDDYYYY. The mixed alpha-numeric month/day/year format. For example, 12/31/2010, 12/31/10 or December 31, 2010.
- 2 - DDMMYYYY. The European day/month/year format. For example, 31/12/2010 or 31/12/10.
- 3 - DDMMMYYYY. The European mixed alpha-numeric day/month/year format. For example, 31/12/2010, 31/12/10, or 31 December 2010.
- 4 - YYYY. The numeric year format. For example, 2010 or 10.
- 5 - MM. The numeric month format. For example, 12.
- 6 - DD. The numeric day format. For example, 31.
- 7 - MYYYYY. The numeric month/year format. For example, 12/2010 or 12/10.
- 8 - MMDD. The numeric month/day format. For example, 12/31.
- 9 - YYYYMMDD. The numeric year/mm/day format. For example, 2010/12/31 or 10/12/31.

- 10 - GenericUS. The combination of date formats generally used in the United States. This is a combination of date formats MMDDYYYY, YYYYMMDD, and MMMMDDYYYY.

Returns

True always.

Level

Field.

Details

Sets the date format for date fields. The date type supports the following languages: English, German, Portuguese (Brazil), or Spanish (Latin America). The date formats DD MMMM YYYY and MMMM DD, YY are not supported for German language.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType (512)
SetDateFormat (1)
```

The example above sets the field type to Date, and sets the expected date format.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetDeskew

Turns on or off the deskew on the field snippet prior to recognition.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetDeskew (string param)
```

Parameters

The boolean value specifying whether or not to perform deskew on the field snippet prior to recognition. The following are the valid values:

- True - Deskew is performed (default).
- False - Deskew is not performed.

Returns

True always.

Level

Field.

Details

This action turns the deskew on or off on the field snippet prior to recognition.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetDeskew("False")
SetFieldType(2)
SetValidValues("1000,2000")
```

The example above turns off deskew on this field, sets the field type to numeric, and finally sets the minimum value to 1000, maximum value to 2000.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetFieldType

Sets the type of the field to read.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetFieldType (string param)
```

Parameters

The integer value representing field type. The following values are supported:

- **1 - Alpha** - Describes a field containing alpha characters. Fields of type Alpha or Alpha numeric are intended to recognize a field that would contain 1, 2 or 3 words. The supported character set includes the letters A to Z (upper, lower, or mixed case) as well as the following special characters:

space; apostrophe; dash; comma; period; slash; pound sign; plus; dollar sign; parentheses; asterisk; quotation marks.

You can use the following actions to improve recognition of this field type:

- SetVocabulary
 - SetWritingStyle
 - SetLanguage
 - SetSpecialCharacterSet
 - SetValidValues
 - SetProcessingMode
- **2 - Numeric** - Describes a field containing numbers. The supported character set includes the numerals 0 to 9 as well as the following special characters:

space; apostrophe; dash; comma; period; slash; pound sign; plus; dollar sign; parentheses; asterisk; quotation marks.

The selected language must match the expected format of the characters. For example, a 7 with a stroke through it is considered European and would recognize better with the language set to French instead of English. The Numeric type supports the languages English, German, Portuguese (Brazil), or Spanish (Latin America).

You can use the following actions to improve recognition of this field type:

- SetVocabulary
 - SetWritingStyle
 - SetLanguage
 - SetSpecialCharacterSet
 - SetValidValues
 - SetProcessingMode
- **4 - Alphanumeric** - Describes a field containing a combination of alpha and numeric characters. This Field Type combines the attributes of an Alpha Field Type and a Numeric Field Type. Fields of type Alpha or Alpha numeric are intended to recognize a field that would contain 1, 2 or 3 words.

Note: You must set a vocabulary or template for this field type. If the vocabulary or template is not set, an error may occur. To set the template, use the SetTemplate or SetVocabulary actions.

You can use the following actions to improve recognition of this field type:

- SetTemplate (*required if SetVocabulary is not called)
 - SetVocabulary (*required if SetTemplate is not called)
 - SetWritingStyle
 - SetLanguage
 - SetSpecialCharacterSet
 - SetValidValues
 - SetProcessingMode
- **8 - Address** - The Address type has the following formats:
 - Address Block
 - Street + City, State, ZIP
 - Street + City
 - State + ZIP
 - Street + City + State + ZIP
 - Street + Apt + City, State, ZIP
 - Street + Apt + City, State + ZIP
 - Street + Apt + City + State + ZIP
 - City, State, ZIP; City, State + ZIP
 - City + State + ZIP

Note: The comma defines the parts of the fields as being joined into one entity. The “+” sign indicates that the parts are separate entities. Only US addresses are supported.

Important: In order to get address recognition results, you must set the postal database path using the SetPostalDatabase action.

When the address is recognized, for each component of the address that is recognized a DCO variable is created in the field object and the variable contains the respective portion of the address. The following DCO variables can be created: Primary, Street, City, State, Zip, ZipFull.

You need to configure the Address field for the following scenarios:

Scenario 1: The address is to be recognized and parsed from a single zone containing the street address (apartment, if present) and city, state, zip. For this scenario, it suffices to create the zone around the entire address and set the field type to address. No additional fields or setup is needed, beyond specification of the postal database. The multiple line mode must be set to **None** via the SetMultiLineMode action (parameter of 0). If this value is not set the results may only contain text for only a single line of the address block. Upon completion of the recognition, the field value is set to the text of entire address and the variables specified above are created containing the respective portions of the address.

Scenario 2: Use this scenario when the address is to be recognized and parsed from multiple zones because it is not written in a standard contiguous format. For example, street address, city, state, and zip are in different zones. For this scenario, the following actions are available to configure the different zones for each part of the address as needed:

SetAddressCityStateZip, SetAddressCityState, SetAddressApartment, SetAddressCity, SetAddressState, SetAddressZip

For this scenario, each of the fields called out in the SetAddress actions above must be zoned on the page area that contains each of the respective portions of the address. For example, the field specified for SetAddressCityState must be zoned to include both the city and state and the field specified for SetAddressCity must be zoned just for the city text. It is not required that all actions and zones are created. If the city and state are written in the contiguous form "city, state", then it makes sense to zone them as one field and use SetAddressCityState to identify the field. If the city and state do not appear as contiguous on the page, then the city and state can be zoned separately using two fields and use the actions SetAddressCity and SetAddressState to identify each field. This scenario uses a set of fields to recognize the address. One main field is configured with SetFieldType(8) and then uses the set actions to identify the other fields that comprise the address to be recognized. The main field zone does not need to be surrounding the entire address as the address is obtained from the identified address parts. When recognition is complete, the text value of each of the fields specified by SetAddress is filled with the recognized text. Additionally the main field that is configured with SetFieldType(8) contains DCO variables mentioned above that contain the respective portions of the recognized address. The field text is not updated with any of the recognized text unless it has been zoned around an address field.

For both the scenarios, you can use the following actions to improve recognition of this field type:

- SetPostalDatabase (*required)
- SetWritingStyle
- **16 - Age** - Describes a field containing age data. This Field Type supports:
 - A numeric value, for example 37.
 - An alpha-numeric representation, for example 14 years old.

The supported value units include the following words and abbreviations:

- Day, Days, Dys

- o Mo, Month, Months, Mos, Mths, Mth, Month Old, Months Old, Mos Old, Mths Old
- o Week, Weeks, Wk, Wks
- o Year, Years, Years Old, Yr, Yrs, Yrs Old

You can use the following actions to improve recognition of this field type:

- o SetWritingStyle
- o SetSpecialCharacterSet
- o SetValidValues
- o SetProcessingMode

- **32 - Amount** - Describes a field containing a dollar amount and/or a cent amount. The supported character set includes numerals as well as the following:

space; dash; comma; period; dollar sign

You can use the following actions to improve recognition of this field type:

- o SetAmountFormat
- o SetWritingStyle
- o SetValidValues
- o SetProcessingMode

- **64 - Barcode** - Not supported. For barcode recognition please use the actions in the [Barcode_P actions](#) library.

- **128 - CheckBox** - Not supported. For check box recognition please use the action `RecogOMRThreshold` in the [Barcode_P actions](#) library.

- **256 - Credit Card Number** - Describes a field containing credit card number.

You can use the following actions to improve recognition of this field type:

- o SetWritingStyle
- o SetValidValues
- o SetProcessingMode
- o SetCreditCardFormat

- **512 - Date** - The date type has the following formats:

MM/DD/YY (MM/DD/YYYY); YYYY; MM; DD; MM/YY; MMMM DD, YYYY; DD/MM/YY (DD/MM/YYYY)

The date type supports the languages English, German, Portuguese (Brazil), or Spanish (Latin America). The date formats DD MMMM YYYY and MMMM DD, YY are not supported for German language.

You can use the following actions to improve recognition of this field type:

- o SetDateFormat
- o SetWritingStyle
- o SetProcessingMode

- **1024 - Empty** - Use to skip recognition of the field. This field type can be used to skip recognition on fields used only to hold zone information for address fields.

- **2048 - First name** - Describes a field that contains first name of a person. Often the first name is written together with the middle initial. In this situation it is still possible to assign a First name type to the field.

You can use the following actions to improve recognition of this field type:

- SetVocabulary
 - SetWritingStyle
 - SetLanguage
 - SetSpecialCharacterSet
 - SetValidValues
 - SetProcessingMode
- **4096 - Full name** - Describes a field on a form that contains first name of a person, middle initial and last name written within one box. It is recommended that you define a field as Full name when it is impossible to reliably separate first name, middle initial, and last name and define them as separate fields with individual field types.

You can use the following actions to improve recognition of this field type:

- SetFullNameFormat
 - SetVocabulary
 - SetWritingStyle
 - SetLanguage
 - SetSpecialCharacterSet
 - SetProcessingMode
- **8192 - Last name** - Describes a field that contains last name of a person.

You can use the following actions to improve recognition of this field type:

- SetVocabulary
 - SetWritingStyle
 - SetLanguage
 - SetSpecialCharacterSet
 - SetProcessingMode
- **16384 - Middle Initial** - Describes a field that contains middle initial of a person.

You can use the following actions to improve recognition of this field type:

- SetVocabulary
 - SetWritingStyle
 - SetLanguage
 - SetSpecialCharacterSet
 - SetProcessingMode
- **32768 - Phone number** - Describes a field containing an area code, a telephone number, or both. The possible format of the field is defined by Field Properties and includes the following:

(000); 000-0000; (000) 000-0000; [(000)] 000-0000

You can use the following actions to improve recognition of this field type:

- SetVocabulary
- SetWritingStyle
- SetLanguage
- SetSpecialCharacterSet
- SetValidValues
- SetProcessingMode

- **65536 - PostNet Barcode** - Not supported. For barcode recognition please use the actions in the [Barcode_P actions](#) library.
- **131072 - Social security number** - Describes a field containing a social security number. The supported character set includes the numerals 0 to 9 as well as the special characters:

space; dash; period; slash

You can use the following actions to improve recognition of this field type:

- SetVocabulary
 - SetWritingStyle
 - SetLanguage
 - SetSpecialCharacterSet
 - SetValidValues
 - SetProcessingMode
- **262144 - Length** - Describes a field containing length measurement expressed in feet, inches, or both. The supported character set includes Numeric and the following special characters:

dash; apostrophe; quotation marks

The supported formats include:

NN' - NN"; NN' NN"; NN NN; NN-NN; NN'NN"

You can use the following actions to improve recognition of this field type:

- SetWritingStyle
 - SetValidValues
 - SetProcessingMode
- **524288 - Measure** - Describes a field containing a numerical value and an abbreviation, for example 115 lbs.. The supported character set includes letters and numbers as well as the following special characters:

period; pound sign

The measurement units (lbs., oz., kg., mm., and so on) should not exceed four characters.

You can use the following actions to improve recognition of this field type:

- SetWritingStyle
 - SetValidValues
 - SetProcessingMode
- Note:
- French language is only supported for the following field types: Alpha, Alpha-numeric, and Numeric for Cursive writing style
 - Russian language is supported for the following field types: Alpha, Alpha-numeric, First Name, Last Name, Middle Initial, and Numeric

Returns

True always.

Level

Field.

Details

Sets the type of the field to read.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldTtype (8)  
SetAddressCityStateZipZone ("fieldCZS")
```

The example above sets the field type to Address. Then sets the zone for the city, state, and zip text in the image. This assumes that the zones for field FieldCSZ are populated previously by an action such as ReadZones.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetFullNameFormat

Sets the format of full name fields.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetFullNameFormat (string param)
```

Parameters

The integer value representing the full name format. The following values are supported:

- 0 - First name followed by last name (default)
- 1 - Last name followed by first name

Returns

True always.

Level

Field.

Details

Sets the format of full name fields.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType (4096)
SetFullNameFormat (1)
```

The example above sets the field type to full name, and sets the full name format to first,last.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetLanguage

Sets the recognition language for the fields.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetLanguage (string param)
```

Parameters

The integer value representing the recognition language. The following values are supported:

- 1 - English (default)
- 2 - German
- 4 - French
- 8 - Russian
- 16 - Brazilian Portuguese
- 32 - Latin American Spanish

Returns

True always.

Level

Field.

Details

Sets the recognition language for the fields.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType (1)  
SetLanguage (2)
```

The example above sets the field type to alpha, and sets the language to German.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetLineRemovalMode

Sets the type of line removal to be performed on the field.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetLineRemovalMode (string param)
```

Parameters

The integer value representing the type of line removal to perform on the field. The following are the valid values:

- 0 - None
- 1 - Vertical lines
- 2 - Horizontal lines
- 3 - Vertical and horizontal lines (default)
- 4 - Vertical and horizontal dash lines

Returns

True always.

Level

Field.

Details

Sets the type of line removal to be performed on the field.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetLineRemovalMode (1)  
SetFieldType (1)  
SetMultiLineMode (2)
```

The example above set line removal to only remove vertical lines, then sets the field type to alpha, and finally sets the recognition to read only the second line.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset)

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetMultiLineMode

Sets which lines to extract from multi-line fields.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetMultiLineMode (string param)
```

Parameters

The integer value representing the way to read the lines, in fields, which have multiple lines. The following are the valid values:

- 0 - None
- 1 - Top line
- 2 - Second line
- 3 - Third line
- 4 - Fourth line
- 5 - Bottom line
- 6 - Second from bottom line
- 7 - Third from bottom line
- 8 - Fourth from bottom line
- 9 - All beneath top
- 10 - All above bottom
- 11 - Default or All above bottom
- 12 - All (default)

Returns

True always.

Level

Field.

Details

Sets which lines to extract from multi-line fields.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType (1)  
SetMultiLineMode (2)
```

The example above sets the field type to alpha, and sets the recognition to read only the second line.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetNoiseRemoval

Turns on or off the noise removal on the field snippet prior to recognition.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetNoiseRemoval (string param)
```

Parameters

A boolean value specifying whether or not to perform noise removal on the field snippet prior to recognition. The following are the valid values:

- True - Noise removal is performed (default)
- False - Noise removal is turned off

Returns

True always.

Level

Field.

Details

This action turns the noise removal on or off on the field snippet prior to recognition.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetNoiseRemoval ("False")  
SetFieldType (2)  
SetValidValues ("1000,2000")
```

The example above turns off noise removal on this field, sets the field type to numeric, and finally sets the minimum value to 1000, maximum value to 2000.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetPostalDatabase

Sets the path of the postal database for address recognition.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetPostalDatabase (string param)
```

Parameters

Smart parameters are supported.

Returns

True always.

Level

Field.

Details

Sets the full path of the postal database file ads_database.cfg, required for address recognition. An address must exist in the postal database for it to be recognized correctly.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType(8)
SetPostalDatabase("c:\PostalDatabase\ads_database.cfg")
```

The example above sets the field type to address, and sets the path to the postal database.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetProcessingMode

Sets whether to favor speed or accuracy when performing recognition.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetProcessingMode (string param)
```

Parameters

The integer value representing whether to favor speed or accuracy when performing recognition. The following are the valid values:

- 0 - Favor Accuracy (slower, default)
- 1 - Speed (Faster)

Returns

True always.

Level

Field.

Details

Sets whether to favor speed or accuracy when performing recognition.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetProcessingMode(1)
SetFieldType(1)
SetMultiLineMode(2)
```

The example above sets recognition to favor speed, then sets the field type to alpha, and finally sets the recognition to read only the second line.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetSpecialCharacterSet

Sets additional characters to add to the default field character set.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetSpecialCharacterSet (string param)
```

Parameters

The additional characters to add to the default field character set.

Returns

True always.

Level

Field.

Details

Sets additional characters to add to the default field character set. If spaces should be included in the results, use an underscore "_" as the character to specify a space. A special character set can be provided for the Alpha, Alphanumeric, and Numeric field types.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType (1)  
SetSpecialCharacterSet ("12345")
```

The example above sets the field type to alpha, and adds the digits 1-5 to the alpha character set to use during recognition. To allow recognition of spaces, use an underscore _ in the special character set.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

SetTemplate

Sets the format of the field text to read.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetTemplate (string param)
```

Parameters

The string value representing the format of the field text.

- # matches numerals
- @ matches alpha characters
- \$ matches special symbols
- * matches any symbol
- | separates multiple formats for the template

None

Returns

True always.

Level

Field.

Details

Sets the format of the field text to read. If the recognized text does not match the template, then no result is returned.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType (4)  
SetTemplate ("@@#####")
```

The example above sets the field type to AlphaNumeric and sets the field template to two alpha characters, followed by four numbers.

```
SetTemplate ("@@##### | ***")
```

The example above allows two alpha followed by four numbers or any three characters. Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

Recognize ()

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetValidLength

Sets the length range for the field.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetValidLength (string param)
```

Parameters

A comma separated value of two integers. The first one representing the minimum length, and the second one representing the maximum length.

Returns

True always.

Level

Field.

Details

Sets the length range for the field.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType(1)  
SetValidLength("3,5")
```

The example above sets the field type to alpha, and sets the minimum length to 3, and maximum length to 5.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetValidValues

Sets the length range for the field.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetValidValues (string param)
```

Parameters

A comma separated value of two integers. The first one representing the minimum value and the second one representing the maximum value.

Returns

True always.

Level

Field.

Details

Sets the length range for the field.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType (2)  
SetValidValues ("1000,2000")
```

The example above sets the field type to numeric, and sets the minimum value to 1000, and maximum value to 2000.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize ()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetVocabulary

Sets the vocabulary file to use for recognizing fields.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetVocabulary (string param)
```

Parameters

A string value representing the vocabulary file to use for recognizing this field.

Returns

True always.

Level

Field.

Details

Sets the vocabulary file to use for recognizing fields. A vocabulary is required when recognizing fields of type AlphaNumeric. You can create vocabularies using the VocabularyEditor.exe in the FormXtra directory. A vocabulary is a list of all of the expected words to be recognized, and weights are assigned to each word. Words that are expected often should be given a higher weight and words expected less often should be given a lower weight.

If a vocabulary is not specified for the AlphaNumeric fields, you must set a template using the SetTemplate action. A vocabulary can provide more accurate results than a template.

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType(4)  
SetVocabulary("c:\temp\vocab.fvx")
```

The example above sets the field type to AlphaNumeric and the vocabulary file to use to c:\temp\vocab.fvx.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

SetWritingStyle

Sets the writing style of the field.

Member of namespace

HandwritingRecognition

Syntax

```
bool SetWritingStyle (string param)
```

Parameters

The integer value representing the writing style of the field. The following values are supported:

- 1 - Cursive - It is only supported for field types: Address, Alpha, First name, Middle name, Last name, and Full name. Setting the writing styles of other field types to cursive may result in an error.
- 2 - Handprint
- 4 - BoxedHandprint (The text is hand-printed within the confines of special boxes or combs)

Returns

True always.

Level

Field.

Details

Sets the writing style of the field. The language support is different for each field type. The following is the list of the languages supported for each field type for the styles Cursive and Handprint:

- **Alpha:** English, French, Russian, Portuguese (Brazil), Spanish (Latin America)
- **Numeric:** English, German, French, Russian, Portuguese (Brazil), Spanish
- **Alpha-numeric:** English, French, Russian
- **Address:** English
- **Age:** English
- **Amount:** English
- **Credit card number:** English, French (Handprint only), Russian, Portuguese (Brazil), Spanish (Latin America)
- **Date:** English, French (Handprint only), Russian, Portuguese (Brazil), Spanish (Latin America)
- **First name:** English, Russian
- **Full name:** English, Russian
- **Last name:** English, Russian
- **Middle name:** English, Russian
- **Phone number:** English
- **Social security number:** English
- **Length:** English
- **Measure:** English

Example:

Field level ruleset (can be run either when entering a field, Open node, or leaving a field, Close node):

```
SetFieldType(1)  
SetWritingStyle(2)
```

The example above sets the field type to alpha, and sets the writing style to handprint.

Page level ruleset (must be run after ALL fields to be recognized have been configured, either at the Close of the page, or in a separate subsequent ruleset).

```
Recognize()
```

This ruleset performs recognition of the configured fields.

Parent topic: [HandwritingRecognition actions](#)

IBMCM actions

Use the IBMCM actions to upload documents into an IBM® Content Manager Connector repository.

The IBM Content Manager Connector actions integrate Datacap applications with the IBM Content Manager Connector repository. You run these actions to access the IBM Content Manager Connector server, set up document attributes and folders on the server, and upload documents to the server for storage.

- [IBMCM_AddPages](#)
Adds new pages to an existing document in the IBM Content Manager repository. The new pages are appended to the existing pages in the IBM Content Manager document.
- [IBMCM_CreateFolder](#)
Creates an IBM Content Manager folder in the parent folder that is based on the specified item type.
- [IBMCM_CreateItem](#)
Creates an IBM Content Manager document that is based on a specified item type.
- [IBMCM_SetAttributeValue](#)
Sets the attribute value on an IBM Content Manager document or folder.
- [IBMCM_CreateChildItem](#)
Creates a child component under the current document that was created by using IBMCM_CreateItem.
- [IBMCM_SetChildAttributeValue](#)
Assigns a child attribute value for a child item (child component attribute value).
- [IBMCM_DeletePages](#)
Deletes pages from an existing document in the IBM Content Manager repository.
- [IBMCM_Logon](#)
Logs on to an IBM Content Manager Server.
- [IBMCM_ReplacePage](#)
Replaces a page in an existing IBM Content Manager document with a new page in the runtime DCO hierarchy.
- [IBMCM_SearchItem](#)
Searches for the existing item in the IBM Content Manager repository.
- [IBMCM_SetSearchOnlyFolderItems](#)
Prepares the IBMCM_SearchItem action to limit the searches to only folder items. You use this action to search folder and non-folder from Datacap into IBM Content Manager.
- [IBMCM_SetMimeType](#)
Sets the mime type for the upload document.
- [IBMCM_SetDestinationFolder](#)
Sets the destination folder for uploading images to the IBM Content Manager repository.
- [IBMCM_SetDestinationFolderEx](#)
Sets the destination folder for uploading image files to IBM Content Manager repository.
- [IBMCM_StoreItemIDinDCO](#)
Stores the Item ID of the most recently created folder or the most recently uploaded IBM Content Manager item into a variable of the current object of the Document Hierarchy.
- [IBMCM_UploadDCO_DOC](#)
Uploads the set of Images files that are associated with the current document object of the Document Hierarchy to the IBM Content Manager Server.
- [IBMCM_UploadDCO_Page](#)
Uploads image files that are associated with the current Page object of the Document Hierarchy to IBM Content Manager.

- [IBMCM_SearchAndDownload](#)
Searches and downloads item content from IBM Content Manager repository.
- [IBMCM_SetSearchAndDownloadCriteria](#)
Sets the search criteria for IBMCM_SearchAndDownload action.
- [IBMCM_SetSearchAndDownloadDirectory](#)
Sets the download directory for SearchAndDownload action.
- [IBMCM_SetSearchAndDownloadMaximum](#)
Sets the maximum number of downloads for SearchAndDownload action.
- [IBMCM_SetSearchAndDownloadSort](#)
Sets the sortBy and sortOrder for SearchAndDownload action.
- [IBMCM_SetSearchAndDownloadStatusAttribute](#)
Sets the status attribute to update after a successful SearchAndDownload action.

Parent topic: [Global actions](#)

IBMCM_AddPages

Adds new pages to an existing document in the IBM® Content Manager repository. The new pages are appended to the existing pages in the IBM Content Manager document.

Syntax

```
bool IBMCM_AddPages(int newPage)
```

Parameters

Int newPage: the new page to add to the IBM Content Manager document.

Parameters

newpage is required when you call the action at the document level.

If newpage=0, all of the new pages in the runtime DCO are added to the existing IBM Content Manager document.

If newpage>0, only the new page that is specified by newpage is added.

Returns

True, if the new page or pages are successfully added. Otherwise, False.

Level

Doc and Page levels.

Details

You must retrieve the existing IBM Content Manager document from the IBM Content Manager repository before you call this action to add new pages to it.

You can use the IBMCM_SearchItem action to retrieve the existing IBM Content Manager document.

Example

```
IBMCM_SearchItem("", "A1001001A14B04B12546D00215")
IBMCM_AddPages (2)
```

This example searches the IBM Content Manager repository for the existing document with the unique ID of A1001001A14B04B12546D00215.

If the document is found, this action adds a second page (TM000002) in the runtime Document DCO to this document.

```
IBMCM_SearchItem("EmployeeID", "14B04B12")
IBMCM_AddPages (0)
```

This example searches the IBM Content Manager repository for the existing document with the unique ID of EmployeeID=14B04B12.

If the document is found, this action adds all of the pages in the runtime Document DCO object to this document.

Parent topic: [IBMCM actions](#)

IBMCM_CreateFolder

Creates an IBM® Content Manager folder in the parent folder that is based on the specified item type.

Syntax

```
bool IBMCM_CreateFolder(string itemType, string attribute, string attributeValue,
bool hasParent)
```

Parameters

String itemType: the item type (classification) that is used to create the IBM Content Manager folder. This parameter is required.

String attribute: the attribute name.

String attributeValue: the unique attribute value or folder ID.

bool hasParent: if True the new folder has a parent folder.

Parameters

If hasParent is False, the new folder is created without a parent folder and the action ignores the attribute and attributeValue parameters.

If hasParent is True, it is expected that this folder is the child of an existing folder and that the attribute and attributeValue parameters specify the parent folder.

If hasParent is True and both attribute and attributeValue are not provided, the parent folder is set to the most recently created folder.

If the folder ID is used in the attributeValue parameter, leave the attribute parameter empty ("").

Smart parameters are supported for the string parameters.

Returns

True, if the folder is successfully created. Otherwise, False. The action fails if the a folder with the specified attribute or ID is not found.

Level

All levels.

Details

Creates a folder in the IBM Content Manager repository. The `IBMCM_SetAttributeValue` action can be called following this action to set the attributes of the newly created folder.

Example:

```
IBMCM_CreateFolder("NOINDEX", "", "", False)
```

This example creates an IBM Content Manager folder that is based on the NOINDEX item type. The new folder has no parent and is placed in the root directory.

```
IBMCM_CreateFolder("NOINDEX", "",  
"A1001001A14B04B12546D00215", True)
```

This example creates an IBM Content Manager folder that is based on the NOINDEX item type. The folder is placed inside the parent folder with the ID "A1001001A14B04B12546D00215".

```
IBMCM_CreateFolder(@MyItemType, @MyAttribute,  
@MyAttributeValue, True)  
IBMCM_SetAttributeValue("name, @BATCHID")
```

This example creates an IBM Content Manager folder that is based on the information that is stored in the smart parameter *@ MyItemType*. This folder is a child that is identified by the values from the batch level smart parameters *@ MyAttribute*, and *@ MyAttributeValue*.

```
IBMCM_CreateFolder(@B.MyItemType, "", "", False)  
IBMCM_SetAttributeValue("Name, MyFolder1")  
IBMCM_CreateFolder(@B.MyItemType, "", "", True)  
IBMCM_SetAttributeValue("Name, MyFolder2")
```

This example creates the folders "MyFolder1" and "MyFolder2" based on the type that is identified by `@B.MyItemType`.

The "MyFolder1" folder has no parent folder. The parent folder of "MyFolder2" is the "MyFolder1" folder.

Parent topic: [IBMCM actions](#)

IBMCM_CreateItem

Creates an IBM® Content Manager document that is based on a specified item type.

Member of namespace

ibmcm

Syntax

```
bool IBMCM_CreateItem(string itemtype)
```

Parameters

itemtype: Creates an IBM Content Manager document that is based on the item type.

Parameters

itemtype: a string value of a valid IBM Content Manager Item Type, for example NOINDEX. An `IBM Content Manager Item Type is equivalent to a Document Class (Index Class).

Smart parameters are supported, such as `@BATCHID`, `@ID`, `@STATUS`, `@TYPE`, `@VALUE`, `@JOBID`, `@JOBNAME`, `@OPERATOR`, `@STATION`, `@TASKID`, and `@TASKNAME`. For more information, refer to the smart parameter documentation.

Returns

True If the document is successfully created. Otherwise, False.

Level

Document or Page level.

Details

Creates an IBM Content Manager based on the item type. This action must be called before the document and page upload actions.

Example:

```
IBMCM_CreateItem("NOINDEX")
IBMCM_SetMimeType("image/tiff")
IBMCM_UploadDC)_DOC()
```

This example creates an IBM Content Manager document that is based on the NOINDEX item type. It sets the mime type of the uploaded document, and then runs the upload action.

```
IBMCM_CreateItem(@P.name)
```

This example creates an IBM Content Manager document that is based on the value that is contained inside the Smart Parameter `@P.name` at the Page level.

Typically, there is one item for each processed document that is represented by a Document object of the Document Hierarchy, or for a processed page that is represented by a Page object.

Parent topic: [IBMCM actions](#)

IBMCM_SetAttributeValue

Sets the attribute value on an IBM® Content Manager document or folder.

Member of namespace

ibmcm

Syntax

```
bool IBMCM_SetAttributeValue (string attributesvalues)
```

Parameters

String *attributesvalues*: Sets the attribute value on an IBM Content Manager document or folder.

Parameters

attributesvalues is a string value of two comma separated variables:

Name: The name of the variable to set.

Value: The value of the variable to set.

A IBM Content Manager Item Type is equivalent to a Document Class (Index Class).

Smart parameters are supported, such as *@BATCHID*, *@ID*, *@STATUS*, *@TYPE*, *@VALUE*, *@JOBID*, *@JOBNAME*, *@OPERATOR*, *@STATION*, *@TASKID*, and *@TASKNAME*. Refer to the smart parameter documentation for more information and usage.

Returns

True if the document is successfully created. Otherwise, False.

Level

All levels.

Details

Sets the attribute value on a IBM Content Manager document or folder. The specified attribute must already be defined on the IBM Content Manager server. The attribute can be applied to a document or a folder. This action must follow a previous action that created the new folder or that identifies a document. This action can be called multiple times to set multiple processes.

Example:

```
IBMCM_CreateFolder("MyItemType")
IBMCM_SetAttributeValue("Name,@BATCHID")
```

This example creates an IBM Content Manager folder and sets the *Name* property of the folder to the ID of the current batch.

```
IBMCM_CreateItem("@B.name")
IBMCM_SetAttributeValue("Author,@D.TheAuthor")
IBMCM_SetAttributeValue("Date,@DATE")
```

When running on the document level, this example creates a new document by using the batch level value that is stored in the DCO variable *Name* and sets the *Author* attribute to the value that is stored in the DCO document level variable named *TheAuthor*. Then it stores the current date in the *Date* property.

```
IBMCM_SearchItem("employeeID","3F1234D")
IBMCM_SetAttributeValue("FirstName,Thomas")
```

This example searches the IBM Content Manager repository for an existing IBM Content Manager document with the EmployeeID that is 3F1234D. If the existing IBM Content Manager document is found, then its attribute *FirstName* is set to *Thomas*.

Parent topic: [IBMCM actions](#)

IBMCM_CreateChildItem

Creates a child component under the current document that was created by using IBMCM_CreateItem.

Member of namespace

ibmcm

Syntax

```
bool IBMCM_CreateChildItem(string attributeInfo)
```

Parameters

attributeInfo: String in the format of *child item name, count*.

Returns

True, if the child item is created is successfully created. Otherwise, False.

Level

Document or Page level.

Details

Creates a child component under the current document that was created by using IBMCM_CreateItem. A comma-separated string that consists of two values:

childItemName

Defines a child component under the IBM® Content Manager item that was created by IBMCM_CreateItem.

The parameter is a string value that specifies the name of a child component.

count

The number of child component attribute values that will be set.

The count is the number of child component attributes values that will be set by IBMCM_SetChildAttributeValue.

This action needs to be called after IBMCM_CreateItem.

Example:

```
IBMCM_CreateItem("ABCInsPlc")
IBMCM_CreateChildItem("Automobiles,1")
IBMCM_SetChildAttributeValue("Automobiles,ABCVIN,vin123")
IBMCM_UploadDCO_DOC()
```

This example creates an IBM Content Manager Child Item based on the name `Automobiles`.

Parent topic: [IBMCM actions](#)

IBMCM_SetChildAttributeValue

Assigns a child attribute value for a child item (child component attribute value).

Member of namespace

ibmcm

Syntax

```
bool IBMCM_SetChildAttributeValue(string attributeInfo)
```

Parameters

attributeInfo: String in the format of *child item name, attribute name, attribute value*.

Returns

True, if the child attribute value is set successfully. Otherwise, False.

Level

All levels.

Details

Assigns a child attribute value for a child item. (child component attribute value)

A comma-separated string that consists of three values:

childItemName

Defines a child component under the IBM® Content Manager item that was created by IBMCM_CreateItem.

The parameter is a string value that specifies the name of a child component.

attributeName

The attribute name of the child.

attributeValue

The attribute value.

IBMCM_CreateChildItem must be called before this action can be called.

IBMCM_SetChildAttributeValue must be called the same number of times that is specified in IBMCM_CreateChildItem otherwise the child attributes will not be saved

Smart parameters are supported.

Example:

```
IBMCM_CreateItem("ABCInsPlc")
IBMCM_CreateChildItem("Automobiles,2")
IBMCM_SetChildAttributeValue("Automobiles,ABCVIN,vin123")
IBMCM_SetChildAttributeValue("Automobiles,ABCLicensePlate,carlic123")
IBMCM_UploadDCO_DOC()
```

This example creates an IBM Content Manager Child Item based on the name `Automobiles` and assigns two attributes.

IBMCM_DeletePages

Deletes pages from an existing document in the IBM® Content Manager repository.

Syntax

```
bool IBMCM_DeletePages(int existingPage)
```

Parameters

Int existingPage: the existing page to delete from the document. This parameter is required.

Parameters

existingPage: required when you call this action at the Document level.

If existingPage = 0, all of the existing pages in the IBM Content Manager document are deleted. Otherwise, only the existing page that is specified by existingPage is deleted from the document.

Returns

True, if the page or pages are successfully deleted. Otherwise, False.

Level

All levels.

Details

You must retrieve the existing IBM Content Manager document from the IBM Content Manager repository before you call this action to delete pages from it.

The IBMCM_SearchItem action can be used to retrieve the existing IBM Content Manager document.

Important: Be careful with this destructive action. It should be used at the Batch level to make sure that it is run one time per batch. If you must use it at the Document or Page level, make sure that you set filters and conditions to keep this action from being called repeatedly.

Example

```
IBMCM_SearchItem("EmployeeID", "14B04B12")  
IBMCM_DeletePages(2)
```

This example searches the IBM Content Manager repository for the existing IBM Content Manager document with the unique attribute EmployeeID=14B0B12.

If the existing IBM Content Manager document is found, it deletes the second page from the existing IBM Content Manager document.

```
IBMCM_SearchItem("EmployeeID", "14B04B12")  
IBMCM_DeletePages(0)
```

This example searches the IBM Content Manager repository for the existing IBM Content Manager document with the unique attribute EmployeeID=14B0B12.

If the existing IBM Content Manager document is found, it deletes all of the pages from the existing IBM Content Manager document.

Parent topic: [IBMCM actions](#)

IBMCM_Logon

Logs on to an IBM® Content Manager Server.

Member of namespace

ibmcm

Syntax

```
bool IBMCM_Logon(string connectioninfo)
```

Parameters

connectioninfo

Type: string

A comma-separated string consisting of three values:

1. The ID of the IBM Content Manager Server
2. A valid Content Manager User ID
3. The user's Password

Smart parameters are supported for each parameter.

Returns

True if the logon succeeds. False if the logon is unsuccessful. The logon is unsuccessful if the action cannot find the specified server, or if the user ID or password is invalid.

Level

Any level. Usually at the Batch level.

Details

Performs the logon to the IBM Content Manager system. This action must be called before the actions that communicate with the IBM Content Manager repository. Connectivity, based on the requirements of the IBM Content Manager Server, must be setup on the machine that runs the Datacap rules.

Example:

```
IBMCM_Logon("ibmcmserver,user1,password")
```

```
IBMCM_Logon("ibmcmserver,user1,@APPVAR(values/adv/MyPassword)")
```

This example uses the smart parameter @APPVAR to get the password from the advanced value section of the Application Manager. The custom value name is "MyPassword".

Parent topic: [IBMCM actions](#)

IBMCM_ReplacePage

Replaces a page in an existing IBM® Content Manager document with a new page in the runtime DCO hierarchy.

Syntax

```
bool IBMCM_ReplacePage(int existingPage, int newPage)
```

Parameters

Int existingPage: the existing page in the IBM Content Manager document.

Int newPage: the new page in the DCO hierarchy.

Parameters

When you call this action at the Document level, both existingPage and newPage parameters are required.

When you call this action at the Page level, the newPage parameter is ignored.

Returns

True, if the existing page in the IBM Content Manager document is replaced with the new page in the runtime DCO. Otherwise, False.

Level

Document and Page level.

Details

You must retrieve the existing IBM Content Manager document from the IBM Content Manager repository before you call this action.

The IBMCM_SearchItem action can be used to retrieve the existing IBM Content Manager document.

Example

```
IBMCM_SearchItem("EmployeeID", "14B04B12")  
IBMCM_ReplacePage(2, 3)
```

This example searches the IBM Content Manager repository for the existing IBM Content Manager document with the unique attribute EmployeeID=14B0B12.

If the existing IBM Content Manager document is found, it replaces the second page in the existing IBM Content Manager document with the third page (TM000003) in the runtime Document DCO.

Parent topic: [IBMCM actions](#)

IBMCM_SearchItem

Searches for the existing item in the IBM® Content Manager repository.

Syntax

```
bool IBMCM_SearchItem (string attribute, string attributeValue)
```

Parameters

String attribute: the attribute name.

String attributeValue: the unique attribute value or folder ID.

Parameters

Both attribute and attributeValue are required parameters. If folder ID is used in the second parameter, leave the first parameter empty ("").

Smart parameters are supported.

Returns

True, if the item is found. Otherwise, False.

Level

All levels.

Details

Searches for the existing item in the IBM Content Manager repository that matches the specified attribute and value. If an item is found, the current item is set to it. Otherwise the current item is set to NULL.

IBMCM_SearchItem is used to retrieve an existing IBM Content Manager item before calling other actions such as IBMCM_AddPages, IBMCM_DeletePages, IBMCM_ReplacePage, and IBMCM_SetAttributeValue to update the attributes of the item or the contents of the item.

Example

```
IBMCM_SearchItem("Department", "Human Resource")
IBMCM_SetAttributeValue("Department", "Operations")
```

This example searches for an item with the attribute name *Department* and the attribute value *Human Resources*. It then changes the attribute value to *Operations*.

```
IBMCM_SearchItem("", "A1001001A14B04B12546D00215")
```

This example searches for an item with the ID equal to *A1001001A14B04B12546D00215*.

Parent topic: [IBMCM actions](#)

IBMCM_SetSearchOnlyFolderItems

Prepares the IBMCM_SearchItem action to limit the searches to only folder items. You use this action to search folder and non-folder from Datacap into IBM® Content Manager.

Syntax

```
bool IBMCM_SetSearchOnlyFolderItems (string strParam)
```

Parameters

String strParam: True or False

A string or smart parameter value is True or False, whether the subsequent IBMCM_SearchItem actions limit the result to folder items.

Smart parameters are supported.

Returns

False, if parameter is not a string. Otherwise, True.

Level

All levels.

Details

If True, the subsequent IBMCM_SearchItem actions search for folder items. If False, the subsequent IBMCM_SearchItem actions search for all items that match the search criteria.

If this action is not called or the parameter is not True or False, the default value of True is used, and IBMCM_SearchItem actions only return folder items. This action should be called before IBMCM_SearchItem.

Example

```
IBMCM_SetSearchOnlyFolderItems ("False")
IBMCM_SearchItem ("", "A1001001A17E08B00210C00000")
```

This example demonstrates the search for an item with item ID A1001001A17E08B00210C00000. If an item is found, regardless of whether it is a folder item or not, the current item is set to it. Otherwise, if the item is not found, the current item is set to NULL.

```
IBMCM_SetSearchOnlyFolderItems ("True")
IBMCM_SearchItem ("", "@P.PageName")
```

This example demonstrates the search only for a folder item. If a folder item is found, the current item is set to it. Otherwise, if the item is not found or the item is not a folder item, the current item is set to NULL.

This example demonstrates the use of the predefined special page variable @PageName. The VScan.Scan action populates @P.PageName to be the file name of the image file without the file extension. For example, the image file A1001001A17E08B00210C00000.tif has a PageName of "A1001001A17E08B00210C00000". This page variable can be used to identify the IBM Content Manager item ID if the image file name conforms to the convention.

Member of object Datacap.Libraries.IBMCM.Actions.

Parent topic: [IBMCM actions](#)

IBMCM_SetMimeType

Sets the mime type for the upload document.

Syntax

```
bool IBMCM_SetMimeType(string mimeType)
```

Parameters

String mimeType: the mime type.

Parameters

Smart parameter is supported.

Returns

True, if the mime type is successfully set. Otherwise, False.

Level

All levels.

Details

Set the mime type for the upload document. This action is needed when you are uploading non-tiff files.

When you call this action at the batch or document level, the same mime type is applied to all of the files in the batch or document DCO.

This action must be called before the document and page upload actions.

Table 1. Supported mime types

MIME Types	File Extensions
application/msword	dod, dot, rtf
application/octet-stream	bin, dms, lha, lzh, exe, class
application/pdf	pdf
application/rtf	rtf
application/lotus-1-2-3	123, wk4, wk3, wk1, wks, wg1
application/lotus-freelance	prz, pre
application/vnd.ms-excel	xls, xlt, xlm, xld, xla, xlc, xlw, xll
application/vnd.ms-powerpoint	ppt, pot, ppa, pps, pwz
application/vnd.visio	vsd
audio/basic	au, snd, ulw
audio/mpeg	mpeg, mpg, m1s, m1a, mp2, mp3, mpm, mpa, mpga
audio/x-aiff	aif, aiff, aifc
audio/x-midi	midi, mid, smf, kar
audio/x-wav	wav

MIME Types	File Extensions
image/bmp	bmp, dib
image/gif	gif
image/jpeg	jpeg, jpg, jpe, jfif, pjpeg, pip
image/tiff	tiff, tif
text/html	html, htm, shtml, plg
text/plain	txt, text
text/xml	xml, dtd
video/mpeg	mpeg, mpg, mpe, m1s, m1v, m1a, m75, m15, mp2
video/quicktime	mov, qt

Example:

```
IBMCM_ SetMimeType ("image/tiff")
```

This example sets the mime type to "image/tiff".

Parent topic: [IBMCM actions](#)

IBMCM_SetDestinationFolder

Sets the destination folder for uploading images to the IBM® Content Manager repository.

Syntax

```
bool IBMCM_SetDestinationFolder(string attribute, string attributeValue)
```

Parameters

String attribute: the attribute name.

String attributeValue: the unique attribute value or folder ID.

Parameters

If both parameters are not provided, the destination folder is set to the most recently created folder.

If the folder ID is used in the second parameter, leave the parameter empty ("").

Smart parameters are supported.

Returns

True, if the destination folder is successfully set. Otherwise, False. The action fails if a folder with the specified attribute or ID is not found.

Level

All levels.

Details

Sets the upload destination folder in the IBM Content Manager repository. To set the destination to a newly created folder, first create the folder by using the `IBMCM_CreateFolder` action. Then, call the `IBMCM_SetDestinationFolder` action with empty ("") parameters.

Example:

```
IBMCM_SetDestinationFolder("Department", "Human Resource")
IBMCM_CreateItem("NOINDEX")
IBMCM_SetMimeType("image/tif")
IBMCM_UploadDCO_DOC()
```

This example sets the destination folder to the folder with the attribute name "Department" and the attribute value "Human Resource" to upload images to it.

```
IBMCM_SetDestinationFolder("", "A1001001A14B04B12546D00215")
IBMCM_CreateItem("NOINDEX")
IBMCM_SetMimeType("image/tif")
IBMCM_UploadDCO_DOC()
```

This example sets the destination folder to folder with the ID "A1001001A14B04B12546D00215" to upload images to it.

```
IBMCM_CreateFolder("NOINDEX", "", "A1001001A14B04B12546D00215", true)
IBMCM_SetDestinationFolder("", "")
IBMCM_CreateItem("NOINDEX")
IBMCM_SetMimeType("image/tif")
IBMCM_UploadDCO_DOC()
```

This example sets the destination folder to the most currently created folder to upload images to it.

Parent topic: [IBMCM actions](#)

IBMCM_SetDestinationFolderEx

Sets the destination folder for uploading image files to IBM® Content Manager repository.

Syntax

```
bool IBMCM_SetDestinationFolderEx (string itemType, string attribute, string
attributeValue)
```

Parameters

string itemType - The item type.

string attribute - The attribute name.

string attributeValue - The unique attribute value or folder ID.

Parameters

If all parameters are not provided then destination folder is set to the most recently created folder.

Returns

True, if the destination folder is successfully set. Otherwise, False. The action fails if a folder with the specified attribute or ID is not found.

Level

All levels.

Details

Sets the upload destination folder in IBM Content Manager repository. To set the destination to a newly created folder, first create the folder using `IBMCM_CreateFolder` and then call `IBMCM_SetDestinationFolder` with empty ("") parameters.

Example

```
IBMCM_SetDestinationFolderEx("NOINDEX", "SOURCE", "Human Resource")
IBMCM_CreateItem("NOINDEX")
IBMCM_SetMimeType("image/tiff")
IBMCM_UploadDCO_DOC()
```

This example demonstrates the setting up the destination folder to the folder with attribute `Department="Human Resource"` to upload images to it.

Example

```
IBMCM_SetDestinationFolder("", "A1001001A14B04B12546D00215")
IBMCM_CreateItem("NOINDEX")
IBMCM_SetMimeType("image/tiff")
IBMCM_UploadDCO_DOC()
```

This example demonstrates the setting up the destination folder to the folder with folder ID=`A1001001A14B04B12546D00215` to upload images to it.

Example

```
IBMCM_CreateFolder("NOINDEX", "", "A1001001A14B04B12546D00215", true)
IBMCM_SetDestinationFolder("", "")
IBMCM_CreateItem("NOINDEX")
IBMCM_SetMimeType("image/tiff")
IBMCM_UploadDCO_DOC()
```

This example demonstrates the setting up the destination folder to the most recently created folder to upload images to it

Parent topic: [IBMCM actions](#)

IBMCM_StoreItemIDinDCO

Stores the Item ID of the most recently created folder or the most recently uploaded IBM® Content Manager item into a variable of the current object of the Document Hierarchy.

Member of namespace

ibmcm

Syntax

```
bool IBMCM_StoreItemIDinDCO(string itemID)
```

Parameters

String itemID: sets the attribute value on an IBM Content Manager document or folder.

Returns

True, if the Item ID is returned successfully. Otherwise, False.

Level

All levels.

Details

Stores the Item ID of the most recently created folder or the most recently uploaded IBM Content Manager document into a variable of the current object of the Document Hierarchy. If the variable does not exist, it is created on the current DCO Hierarchy object.

It might be useful to store the item ID if the object is referenced in the following action, such as setting the upload directory.

Example:

```
IBMCM_CreateFolder("NOINDEX", "", "", False)
IBMCM_StoreItemIDinDCO("ItemID")
```

This example stores the ID of the new IBM Content Manager folder in a variable that is called *ItemID* in the current object of the DCO Hierarchy.

```
IBMCM_CreateItem("NOINDEX")
IBMCM_MimeType("image/tiff")
IBMCM_UploadDCO_DOC()
IBMCM_StoreItemIDinDCO("ItemID")
```

This example stores the ID of the new IBM Content Manager document in a variable that is called *ItemID* in the current object of the DCO Hierarchy.

Parent topic: [IBMCM actions](#)

IBMCM_UploadDCO_DOC

Uploads the set of Images files that are associated with the current document object of the Document Hierarchy to the IBM® Content Manager Server.

Member of namespace

ibmcm

Syntax

```
bool IBMCM_UploadDCO_DOC ()
```

Parameters

None

Returns

True if the document is successfully created. Otherwise, False.

Level

Document level

Details

Uploads the Image files that are associated with current Document object of the Document Hierarchy to IBM Content Manager Server.

Example:

```
IBMCM_Logon("cmserver,adminPWD,adminUID")
IBMCM_CreateItem("NOINDEX")
IBMCM_SetAttributeValue("USERID, @OPERATOR")
IBMCM_SetMimeType("image/tiff")
IBMCM_UploadDCO_DOC()
```

This sequence uploads the set of Images files that are associated with the current document object of the Document Hierarchy to the IBM Content Manager server. Additionally, all attributes set by using IBMCM_SetAttributeValue action are also persisted.

Parent topic: [IBMCM actions](#)

IBMCM_UploadDCO_Page

Uploads image files that are associated with the current Page object of the Document Hierarchy to IBM® Content Manager.

Member of namespace

ibmcm

Syntax

```
bool IBMCM_UploadDCO_Page ()
```

Returns

True if the action successfully sends the image to the server. False if the action is unable to save the image.

Level

Page level

Details

Uploads the Image file associated with current Page object of the Document Hierarchy to the IBM Content Manager server. Additionally, all attributes set using IBMCM_SetAttributeValue are also persisted.

Example:

```
IBMCM_Logon (cmserver, adminPWD, adminUID)
IBMCM_CreateItem (NOINDEX)
IBMCM_SetAttributeValue (USERID, @OPERATOR)
IBMCM_UploadDCO_Page ()
```

This sequence uploads Image files associated with the current Page object of the Document Hierarchy to IBM Content Manager, and assigns the name of the object – the value of its Type property.

Parent topic: [IBMCM actions](#)

IBMCM_SearchAndDownload

Searches and downloads item content from IBM® Content Manager repository.

Syntax

```
bool IBMCM_SearchAndDownload ()
```

Returns

True, if one or more items are downloaded. False, if zero items were downloaded or if any errors occurred during parameter validation or searching or downloading.

Level

All levels.

Details

The parameters are used to search for items in IBM Content Manager repository that matches the specified search criteria from the IBMCM_SetSearchAndDownloadCriteria action. For each matching item found, if the corresponding ICMBase parts have content, the content file is written to the directory set from IBMCM_SetSearchAndDownloadDirectory. The output file name is the IBM Content Manager item ID, and the file extension is the same as the original file extension.

To limit the number of downloads for each IBMCM_SearchAndDownload action, you can set a limit with IBMCM_SetSearchAndDownloadMaximum action. The default maximum is 1000.

To change the default sorting, call IBMCM_SetSearchAndDownloadSort.

See the related actions for more details and examples:

- IBMCM_SetSearchAndDownloadCriteria
- IBMCM_SetSearchAndDownloadDirectory
- IBMCM_SetSearchAndDownloadMaximum
- IBMCM_SetSearchAndDownloadSort

Example

```
IBMCM_SetSearchAndDownloadCriteria ("APT", , , ,)
IBMCM_SetSearchAndDownloadMaximum ("100")
IBMCM_SetSearchAndDownloadDirectory ("C:\datacap\APT\images\input")
IBMCM_SetSearchAndDownloadSort ("True", "Date", "ASCENDING")
IBMCM_SearchAndDownload ()
```

This example demonstrates the search and download up to 100 APT IBM Content Manager items' content to the directory, C:\datacap\APT\images\input. The IBM Content Manager query syntax is "/APT SORTBY(@Date ASC)". Each item content has the item ID as the file name. For example, A1001001A17E08B00210C00000.tif .

Member of object Datacap.Libraries.IBMCM.Actions.

Parent topic: [IBMCM actions](#)

IBMCM_SetSearchAndDownloadCriteria

Sets the search criteria for IBMCM_SearchAndDownload action.

Syntax

```
bool IBMCM_SetSearchAndDownloadCriteria (string itemType , string attributeName  
, string comparisonOperator , string attributeValue , string dataType)
```

Parameters

string itemType - Name of the item type

string attributeName - Name of the attribute

string comparisonOperator - Comparison operator

string attributeValue - Comparison value of the attribute

string dataType - Data type of the attribute

Parameters

Item type name is mandatory. The search is limited to the specified item type. The rest of the parameters are optional. If they are not provided, then the search returns all items from the specified item type.

If the attribute name is specified, the item attribute must exist in the IBM® Content Manager repository schema.

If the comparison operator is not specified, but the attribute name and value are specified, then the comparisonOperator defaults to the equal operator, "=". Supported comparison operators are: "=", "<", "<=", ">", ">=", "!=", "LIKE", "NOT LIKE", "IS NULL", "IS NOT NULL", "IN", "NOT IN".

The **comparison value of the attribute** is mandatory if the **attribute name** is specified. For "IS NULL" and "IS NOT NULL" operations, the attribute value and the data type are ignored. For "LIKE" and "NOT LIKE" operations, the attribute value should include the wildcard character "%". For "IN" and "NOT IN" operations, the attribute value should be enclosed in parentheses "()" and delimited by commas. See examples below.

The **data type of the attribute** is optional. Supported data types are: "STRING", "DATE", "NUMBER". The default is "STRING".

Smart parameters are supported.

Returns

False, if parameters are not strings or parameter validations failed. Otherwise, True. Some parameter validations are done in this action, others are done in the IBMCM_SearchAndDownload() action

Level

All levels.

Details

The parameters are used in `IBMCM_SearchAndDownload` action to construct the IBM Content Manager query.

If this action is not called and the search criteria has not been set before `IBMCM_SearchAndDownload()` that returns `False`. This action should be called before `IBMCM_SearchAndDownload`.

Example

```
IBMCM_SetSearchAndDownloadCriteria("APT",,,,)  
IBMCM_SetSearchAndDownloadMaximum("100")  
IBMCM_SetSearchAndDownloadDirectory("C:\datacap\APT\images\input")  
  IBMCM_SetSearchAndDownloadSort("True","ITEMID","ASCENDING")  
IBMCM_SearchAndDownload()
```

This example demonstrates the search and download up to 100 APT IBM Content Manager items' content to the directory, `C:\datacap\APT\images\input`, sorted by the IBM Content Manager item ID. The IBM Content Manager query syntax is `/APT SORTBY(@ITEMID,ASC)`. Each item content has the item ID as the file name. For example, `A1001001A17E08B00210C00000.tif`.

Example

```
IBMCM_SetSearchAndDownloadCriteria("APT", "DCStatus", "=", "NA", )  
IBMCM_SetSearchAndDownloadDirectory("@APPPATH(vscanimagedir) ")  
IBMCM_SetSearchAndDownloadSort("False", , )  
IBMCM_SearchAndDownload()
```

This example demonstrates the search and download up to the default maximum number of APT IBM Content Manager items' content that have not been processed by Datacap into the Smart Parameter VScan image directory. The IBM Content Manager query syntax is `/APT[@DCStatus="NA"]`. Each item content has the item ID as the file name. The file extension is the same as the file extension of the original file. For example, `A1001001A17E08B00210C00000.tif`

Example

```
IBMCM_SetSearchAndDownloadCriteria("APT", "Quantity", ">", "5", "NUMBER")  
IBMCM_SetSearchAndDownloadCriteria("APT", "DCStatus", "IS NULL", , )  
IBMCM_SetSearchAndDownloadCriteria("APT", "DCStatus", "LIKE", "start%", )  
IBMCM_SetSearchAndDownloadCriteria("APT", "DCStatus", "IN", "  
("started", "stopped")", , )  
IBMCM_SetSearchAndDownloadCriteria("APT", "Date", ">", "2017-01-01", "DATE")
```

These are usage examples of the supported data type and comparison operators.

Member of object `Datacap.Libraries.IBMCM.Actions`

Parent topic: [IBMCM actions](#)

IBMCM_SetSearchAndDownloadDirectory

Sets the download directory for `SearchAndDownload` action.

Syntax

```
bool IBMCM_SetSearchAndDownloadDirectory (string path)
```

Parameters

string path - The directory where the downloaded files reside

Parameters

The parameter is used as the download directory in IBMCM_SearchAndDownload action. The output directory is mandatory. You must verify that the directory is valid and has write permissions. Smart parameters are supported.

Returns

True, if parameter is a string. Otherwise, False. The directory is not verified until the IBMCM_SearchAndDownload attempts to write to the directory

Level

All levels.

Details

If this action is not called and the download directory has not been set before IBMCM_SearchAndDownload() that returns False. This action must be called before IBMCM_SearchAndDownload to set the output directory.

Example

```
IBMCM_SetSearchAndDownloadCriteria("APT",,,)
IBMCM_SetSearchAndDownloadMaximum("100")
IBMCM_SetSearchAndDownloadDirectory("C:\datacap\APT\images\input")
IBMCM_SetSearchAndDownloadSort("True","ITEMID","ASCENDING")
IBMCM_SearchAndDownload()
```

This example demonstrates the search and download up to 100 APT IBM® Content Manager items' content to the directory, C:\datacap\APT\images\input. The IBM Content Manager query syntax is "/APT SORTBY(@ITEMID,ASC)". Each item content has the item ID as the file name. For example, A1001001A17E08B00210C00000.tif.

Example

```
IBMCM_SetSearchAndDownloadCriteria("APT", "DCStatus","=","NA")
IBMCM_SetSearchAndDownloadDirectory("@APPPATH(vscanimagedir)")
IBMCM_SetSearchAndDownloadSort("True","Date","ASCENDING")
IBMCM_SearchAndDownload()
```

This example demonstrates the search and download up to the default maximum number of APT IBM Content Manager items' content that have not been processed by Datacap into the Smart Parameter VScan image directory. The IBM Content Manager query syntax is "/APT[@DCStatus='NA'] SORTBY(@Date,ASC)". Each item content has the item ID as the file name. The file extension is the same as the file extension of the original file. For example, A1001001A17E08B00210C00000.tif

Member of object Datacap.Libraries.IBMCM.Actions

Parent topic: [IBMCM actions](#)

IBMCM_SetSearchAndDownloadMaximum

Sets the maximum number of downloads for SearchAndDownload action.

Syntax

```
bool IBMCM_SetSearchAndDownloadMaximum (string strParm)
```

Parameters

string strParm - The maximum number of downloads

Parameters

The strParm sets the maximum number of downloads in the IBMCM_SearchAndDownload action. Smart parameters are supported.

Returns

True, if parameter is a number. Otherwise, False.

Level

All levels.

Details

If this action is not called and the download maximum has not been set before IBMCM_SearchAndDownload() that uses the default maximum, which is 1000. Specify a number less than 1 to have no maximum, and IBMCM_SearchAndDownload() returns all matching items. This action should be called before IBMCM_SearchAndDownload to change the maximum download.

Example

```
IBMCM_SetSearchAndDownloadCriteria("APT",,,)
IBMCM_SetSearchAndDownloadMaximum("100")
IBMCM_SetSearchAndDownloadDirectory("C:\datacap\APT\images\input")
IBMCM_SearchAndDownload()
```

This example demonstrates the search and download up to 100 APT IBM® Content Manager items' content to the directory, C:\datacap\APT\images\input.

Example

```
IBMCM_SearchAndDownload()
```

Since this action was not called, the default maximum is used. This example demonstrates the search and download up to 1000 of content matching the IBMCM_SetSearchAndDownloadCriteria parameters.

Example

```
IBMCM_SetSearchAndDownloadMaximum("0")
IBMCM_SearchAndDownload()
```

Since the maximum is zero, there is no limit to the search. This example demonstrates the search and download all content matching the IBMCM_SetSearchAndDownloadCriteria parameters.

Member of object Datacap.Libraries.IBMCM.Actions

Parent topic: [IBMCM actions](#)

IBMCM_SetSearchAndDownloadSort

Sets the sortBy and sortOrder for SearchAndDownload action.

Syntax

```
bool IBMCM_SetSearchAndDownloadSort (string doSort , string sortBy , string
sortOrder)
```

Parameters

string doSort - True or False to sort

string sortBy - Item attribute name to sort by

string sortOrder - Sort order ASCENDING or DESCENDING

Parameters

The doSort parameter takes True or False and determines whether to sort the search results during the IBMCM_SearchAndDownload action. The sortBy parameter is the name of the item attribute to sort by. The item attribute must exist in the IBM® Content Manager repository schema. The sortOrder parameter takes "ASCENDING" or "DESCENDING" and determines the sort order of the sort by attribute. Smart parameters are supported.

Returns

False, if parameters are missing or invalid. Otherwise, True.

Level

All levels.

Details

If this action is not called and the sort information has not been set before IBMCM_SearchAndDownload() that does not sort the returned results.

This action should be called before IBMCM_SearchAndDownload to change the sort information.

Example

```
IBMCM_SetSearchAndDownloadCriteria("APT","Date","IS NULL",)
IBMCM_SetSearchAndDownloadMaximum("100")
IBMCM_SetSearchAndDownloadDirectory("C:\datacap\APT\images\input")
IBMCM_SetSearchAndDownloadSort("True","Date","ASCENDING")
IBMCM_SearchAndDownload()
```

This example demonstrates the search and download up to 100 APT IBM Content Manager items' content to the directory, C:\datacap\APT\images\input, sorted by the "Date" attribute in ascending order.

Example

```
IBMCM_SearchAndDownload()
```

Since this action was not called, no sorting is done. This example demonstrates the search and download content matching the `IBMCM_SetSearchAndDownloadCriteria` parameters.

Member of object `Datacap.Libraries.IBMCM.Actions`

Parent topic: [IBMCM actions](#)

IBMCM_SetSearchAndDownloadStatusAttribute

Sets the status attribute to update after a successful `SearchAndDownload` action.

Member of namespace

ibmcm

Syntax

```
bool IBMCM_SetSearchAndDownloadStatusAttribute (string checkStatus, string  
attributeName) ()
```

Parameters

string `checkStatus`: True or False whether to add checking for download status.

string `attributeName`: The Datacap download status attribute name.

The `checkStatus` parameter takes True or False and determines whether to check the download status of an item during the `IBMCM_SearchAndDownload` action.

The attribute must exist and should be defined for the `SearchAndDownload` item type as a short integer, with no default. The attribute name is case-sensitive and should be the same as defined in the IBM® Content Manager System Administration Client.

The default behavior is `IBMCM_SetSearchAndDownloadStatusAttribute(False)`. Setting a status attribute is not required for the `IBMCM_SearchAndDownload` action.

Smart parameters are supported.

Returns

True, if parameters are string. Otherwise, False. The attribute is not verified until the `IBMCM_SearchAndDownload` attempts to update the status attribute value.

Level

All level.

Details

The status attribute is used two times during the `IBMCM_SearchAndDownload` action. First, the search criteria is appended to search for the attribute being 0 or NULL. Second, when an item has been downloaded, the

attribute is updated to 1.

Having a status attribute for the download is beneficial to avoid downloading duplicates with the same search criteria between batches.

Example:

```
IBMCM_SetSearchAndDownloadCriteria("APT",,,,)  
IBMCM_SetSearchAndDownloadMaximum("100")  
IBMCM_SetSearchAndDownloadDirectory("C:\datacap\APT\images\input")  
IBMCM_SetSearchAndDownloadStatusAttribute("True", "DCStatus")  
IBMCM_SearchAndDownload()
```

In this example, the IBM Content Manager query syntax is `"/APT[ICMPARTS AND (@DCStatus IS NULL OR @DCStatus=0)]"`. During each batch, 100 items are downloaded and their attributes changed to `@DCStatus=1`. Subsequent batches using the `IBMCM_SearchAndDownload` action does not return any items that have already been downloaded, since the `DCStatus` is no longer null.

Example:

```
IBMCM_SetSearchAndDownloadCriteria("APT", "Date", " < ", "2017-01-01", "DATE")  
IBMCM_SetSearchAndDownloadStatusAttribute("True", "DCStatus")  
IBMCM_SearchAndDownload()
```

In this example, the IBM Content Manager query syntax is `"/APT [ICMPARTS AND @Date < "2017-01-01" AND (@DCStatus IS NULL OR @DCStatus=0)]"`.

Example:

```
IBMCM_SetSearchAndDownloadCriteria("APT", "APT_Date", " < ", "2017-01-01", "DATE")  
IBMCM_SetSearchAndDownloadStatusAttribute("False",)  
IBMCM_SearchAndDownload()
```

In this example, the IBM Content Manager query syntax is `"/APT[ICMPARTS AND @APT_Date < "2017-01-01-01"]"`. No checking is done whether the item has been downloaded by Datacap before. It is recommend that the application calls `IBMCM_SetAttributeValue()` to update the `@APT_Date` value to prevent the item from matching the criteria next time.

Member of object `Datacap.Libraries.IBMCM.Actions`.

Parent topic: [IBMCM actions](#)

ICR_C actions

Use the `ICR_C` actions to recognize constrained (unconnected) hand or computer printed characters. These actions use the OpenText RecoStar engine.

The `ICR_C` actions can recognize characters in fields, page, and zones that are configured for `ICR_C` recognition and store the results in a PDF file.

- [EnableLoggingICR_C](#)
Enables or disables event logging for the ICR/C engine.
- [RecognizeFieldICR_C](#)
Performs character recognition for a specific field.
- [RecognizeFieldVoteICR_C](#)
Recognizes characters based on results from two recognition engines.
- [RecognizePageFields2CCO_ICR_C](#)
Performs recognition for all of the zoned fields on a page.

- [RecognizePageFieldsICR_C](#)
Performs recognition on all fields that have been configured for ICR/C in Datacap Studio.
- [RecognizePageFieldsICR_CEx](#)
Recognizes all fields on the page that have been configured for ICR/C recognition.
- [RecognizePageICR_C](#)
Performs full page recognition using the ICR/C Engine.
- [RecognizePageToPDFICR_C](#)
Performs recognition on the current page and places the results in a PDF file.

Parent topic: [Global actions](#)

EnableLoggingICR_C

Enables or disables event logging for the ICR/C engine.

Member of namespace

icr_c

Syntax

```
bool EnableLoggingICR_C (strParam)
```

Parameters

A boolean value. True enables ICRC logging and False disables ICRC logging. If no parameter is passed in, it uses the default value of True.

Returns

True if the logging state is successfully changed. Otherwise, False.

Level

All levels.

Details

This action enables or disables event logging for the ICR/C engine. Logs are written to the System event log under the entry Datacap.Recognition.Recostar. This action is intended for debugging ICRC recognition problems. If this action is never called, no logging will occur.

Example:

```
EnableLoggingICR_C("true")
```

Parent topic: [ICR_C actions](#)

RecognizeFieldICR_C

Performs character recognition for a specific field.

Member of namespace

icr_c

Syntax

```
bool RecognizeFieldICR_C ()
```

Parameters

None.

Returns

False, if called at the wrong level or if there is a failure in the recognition process. Otherwise, True.

Level

Field level.

Details

Recognition is performed on the current field. The field must have the proper settings in the ICR/C tab in Datacap Studio. This field-level action recognizes characters based on the settings in the ICR/C tab in Datacap Studio and the value is stored in the current field object.

Important: When recognizing an area that contains hand printed characters only, a country, not language, should be selected from the Country dropdown in the ICR/C tab of Datacap Studio for optimal recognition results. If the zoned area contains both hand printed and machine printed characters, then a language, and not a country, should be selected from the Country dropdown in the ICR/C tab of Datacap Studio for optimal recognition results.

Example:

```
RecognizeFieldICR_C()
```

Parent topic: [ICR_C actions](#)

RecognizeFieldVoteICR_C

Recognizes characters based on results from two recognition engines.

Member of namespace

icr_c

Syntax

```
bool RecognizeFieldVoteICR_C ()
```

Parameters

None.

Returns

False, if called at the wrong level or if there is a failure in the recognition process. Otherwise, True.

Level

Field level.

Details

Voting adjusts the assigned confidence of recognized characters based on the results from two different recognition engines. This action is expected to be called after field level recognition is performed on the same field by a different engine.

When this action stores the results of recognition, it first determines if the corresponding Field object of the Document Hierarchy already contains a value. If a value is present, indicating recognition had previously been performed on the field, the action compares the existing value with the field's new recognition results - character by character. If a character's values match the existing value, the Confidence Rating for the character is raised to the maximum level.

Note that when using voting actions, the recognition results are never assigned to the field. Instead, the action changes the Confidence Ratings on the basis of results provided by the first Recognition engine. However, if there are no previous recognition results in the field when this action is called, it will perform like the RecognizeFieldICR_C action.

Example:

```
RecognizeFieldOCR_S()  
RecognizeFieldVoteICR_C()
```

This example first uses the OCR/S engine to recognize the text and store it in the field object. Then it votes on the confidence of the character by comparing it to the ICR/C engine result.

Parent topic: [ICR_C actions](#)

RecognizePageFields2CCO_ICR_C

Performs recognition for all of the zoned fields on a page.

Member of namespace

icr_c

Syntax

```
bool RecognizePageFields2CCO_ICR_C ()
```

Parameters

None.

Returns

False if called at the wrong level or if the CCO does not already exist. Otherwise, True.

Level

Page level.

Details

This action recognizes all of the zoned fields on a page and puts the recognition results for each field in both the Datacap Object Hierarchy and in an already existing CCO. To create a CCO, call either a Full Page OCR action or the AnalyzeImage action before this action.

Example:

```
AnalyzeImage ()  
RecognizePageFields2CCO_ICR_C ()
```

Parent topic: [ICR_C actions](#)

Related reference:

[AnalyzeImage](#)

[Cco2cco actions](#)

RecognizePageFieldsICR_C

Performs recognition on all fields that have been configured for ICR/C in Datacap Studio.

Member of namespace

icr_c

Syntax

```
bool RecognizePageFieldsICR_C ()
```

Parameters

None.

Returns

False, if called at the wrong level or if there is a failure in the recognition process. Otherwise, True.

Level

Page level.

Details

This page-level action recognizes all fields on the page that have been configured for ICR/C recognition in Datacap Studio. Individual field-level recognition actions will overwrite the results from this page-level action. This action will not recognize a zoned field if the Skip Recognition checkbox is selected in the ICR/C tab in Datacap Studio.

Important: The recognition settings must be set individually for each field being recognized. When recognizing an area that contains hand printed characters only, a country, not language, should be selected from the Country dropdown in the ICR/C tab of Datacap Studio for optimal recognition results. If the zoned area contains

both hand printed and machine printed characters, then a language, and not a country, should be selected from the Country dropdown in the ICR/C tab of Datacap Studio for optimal recognition results.

Example:

```
AnalyzeImage ()  
RotateImage ()  
RecognizePageFieldsICR_C ()
```

Parent topic: [ICR_C actions](#)

RecognizePageFieldsICR_CEx

Recognizes all fields on the page that have been configured for ICR/C recognition.

Member of namespace

icr_c

Syntax

```
bool RecognizePageFieldsICR_CEx ()
```

Parameters

None.

Returns

False, if called at the wrong level or if there is a failure in the recognition process. Otherwise, True.

Level

Page level.

Details

This page-level action recognizes all fields on the page that have been configured for ICR/C recognition. Individual field-level recognition actions will overwrite the results from this page-level action. This action will not recognize a zoned field if the Skip Recognition checkbox is selected in the ICR/C tab in Datacap Studio. Important: The recognition settings must be set individually for each field being recognized. When recognizing an area that contains hand printed characters only, a country, not language, should be selected from the Country dropdown in the ICR/C tab of Datacap Studio for optimal recognition results. If the zoned area contains both hand printed and machine printed characters, then a language, and not a country, should be selected from the Country dropdown in the ICR/C tab of Datacap Studio for optimal recognition results.

Example:

```
RecognizePageFieldsICR_CEx ()
```

Parent topic: [ICR_C actions](#)

RecognizePageICR_C

Performs full page recognition using the ICR/C Engine.

Member of namespace

icr_c

Syntax

```
bool RecognizePageICR_C ()
```

Parameters

None.

Returns

False, if called at the wrong level or if there is a failure in the recognition process. Otherwise, True.

Level

Page level.

Details

Performs full page character recognition on the current page based on the settings in the ICR/C tab in Datacap Studio. The action will recognize all characters on the page, and populate the page's CCO file with the recognition results. If a CCO file does not exist at the time this action is called, the action will create one.

Example:

```
AnalyzeImage ()  
RotateImage ()  
RecognizePageICR_C ()
```

This sequence creates a CCO file and checks to see if rotation of the image is needed. Full-page recognition then takes place based on the settings in the ICR/C tab in Datacap Studio. The recognition results are stored in the page's CCO file.

Parent topic: [ICR_C actions](#)

RecognizePageToPDFICR_C

Performs recognition on the current page and places the results in a PDF file.

Member of namespace

icr_c

Syntax

```
bool RecognizePageToPDFICR_C ()
```

Parameters

None.

Returns

False, if called at the wrong level or if there is a failure in the recognition process. Otherwise, True.

Level

Page level.

Details

Performs full page character recognition on the current page based on the settings in the ICR/C tab in Datacap Studio. The action will create a PDF that contains the original page image and the recognized text.

Example:

```
AnalyzeImage ()  
RotateImage ()  
RecognizePageToPDFICR_C ()
```

Parent topic: [ICR_C actions](#)

ImageConvert actions

Use the ImageConvert actions to combine image files or to convert image files to JPEG or TIFF.

The ImageConvert actions can combine multiple pages into a single document, specify both the compression format for file conversion as well as the luminance and color of the output.

- [AppendAllImages](#)
Appends all of the images in the document to the first page.
- [AppendAllImages_ByType](#)
Appends all of the images of a specific type within a document.
- [AppendImage](#)
Concatenates the current image to the bottom of an existing image.
- [AppendImage_StartAsNew](#)
Sets the current page as the first page for a concatenated file.
- [ConvertToJPEG](#)
Converts the current image to a JPEG.
- [ConvertToTIFF](#)
Converts the current image to a TIFF.
- [RescaleImage](#)
Adjusts image for the current page to the specified size.
- [SetChrominanceFactor](#)
Set Compression for JPEG Output Files.
- [SetDeleteOriginal](#)
Controls deletion of file after conversion.
- [SetGrayScale](#)
Controls the grayscale output for JPEG images.
- [SetImageDPIByWidth](#)
Adjusts the DPI of the image and resizes the image based on the expected physical size.
- [SetLuminanceFactor](#)
Sets the image luminance or grayscale quality.

- [SetTIFFCompression](#)
Controls the compression format when saving a TIFF.

Parent topic: [Global actions](#)

AppendAllImages

Appends all of the images in the document to the first page.

Syntax

```
bool AppendAllImages ()
```

Parameters

None.

Returns

True, if all of the TIFF images within the document have been appended to the end of the first TIFF image. Otherwise, False.

Level

Document level.

Details

This action will append (concatenate) all of the images within the document to the first image of the document, creating one long image. The result of the action will modify the first image in the document to become a single continuous image like this:

- Page 1
- Page 2
- Page 3

If images are appended prior to recognition, the entire page area will be recognized and click-n-key enabled.

It is highly recommended that this action be used only documents that have a small number of pages, such as two or three pages per document. The size of the final composite image can quickly become larger than can be handled by some image viewers and possibly subsequent actions. To keep memory usage to a minimum, it is also recommended that 1-bit Black and White images are used.

Note: This action only operates on TIFF images.

Example:

```
AppendAllImages ()
```

Parent topic: [ImageConvert actions](#)

AppendAllImages_ByType

Appends all of the images of a specific type within a document.

Syntax

```
bool AppendAllImages_ByType (StrParam)
```

Parameters

A string that matches the *Type* variable of the page.

Returns

True, if all of the TIFF images that match the specified type within the document have been appended to the end of the first matching TIFF image. Otherwise, False.

Level

Document level.

Details

This action will append (concatenate) all of the images, that are of the type specified by the input parameter, that are all within the same document to the first image of that type, creating one long image. Images are not appended across documents. Assuming you have a document where the first three pages are all the same matching type, the result of the action will modify the first image in the document to become a single continuous image like this:

- Page 1
- Page 2
- Page 3

If images are appended prior to recognition, the entire page area will be recognized and click-n-key enabled.

It is highly recommended that this action be used only documents that have a small number of pages, such as two or three pages per document. The size of the final composite image can quickly become larger than can be handled by some image viewers and possibly by subsequent actions. To keep memory usage to a minimum, it is also recommended that 1-bit Black and White images are used.

Note: This action only operates on TIFF images.

Example:

```
AppendAllImages_ByType ("PO")
```

This example will append all of the images within a document that have a page type of PO.

Parent topic: [ImageConvert actions](#)

AppendImage

Concatenates the current image to the bottom of an existing image.

Syntax

```
bool AppendImage ()
```

Parameters

None.

Returns

True, if the current page is successfully concatenated with the previous page. Otherwise, False.

Level

Page level.

Details

This action concatenates the image for the current page to the bottom of the previous image that is processed by a call to `AppendImage_StartAsNew` or `AppendImage`. If `AppendImage_StartAsNew` is not called, then the first image that is encountered is treated as the starting image.

If images are appended before recognition, the entire page area is recognized and click-n-key enabled.

The image must be a TIFF. See `AppendImage_StartAsNew` for details.

Example:

```
AppendImage ()
```

Parent topic: [ImageConvert actions](#)

Related reference:

[AppendImage_StartAsNew](#)

AppendImage_StartAsNew

Sets the current page as the first page for a concatenated file.

Syntax

```
bool AppendImage_StartAsNew ()
```

Parameters

None.

Returns

True, if the image file exists for the current page and if it is a TIFF file. Otherwise, False.

Level

Page level.

Details

Sets the current page as the start of a new concatenated document. This action is used with `AppendImage` to create a single image that is created by concatenating images to the bottom of the first image. `AppendImage_StartAsNew` identifies the first page and `AppendImage` identifies all subsequent pages. The result of the action modifies the first image in the document to become a single continuous image like this:

- Page 1
- Page 2
- Page 3

If images are appended before recognition, the entire page area is recognized and click-n-key enabled.

It is recommended to use this action only for documents that have a few pages, such as two or three pages per document. The size of the final composite image can quickly become larger than can be handled by some image viewers and possibly subsequent actions. To keep memory usage to a minimum, you can use 1-bit Black and White images.

Attention: This action operates only on TIFF images.

Example:

```
AppendImage_StartAsNew()
```

Parent topic: [ImageConvert actions](#)

Related reference:

[AppendImage](#)

ConvertToJPEG

Converts the current image to a JPEG.

Syntax

```
bool ConvertToJPEG ()
```

Parameters

None.

Returns

True, if the image is converted to a JPEG. False, if the input image is a JPEG or if a failure occurs during the conversion.

Level

Page level.

Details

Converts the current image to a JPEG. The output file name will have a JPG extension.

Supported input file formats: BMP (1, 4, 8, or 24-bit), GIF (1, 4 or 8-bit), PNG (1, 4, 8 and 24-bit), and TIFF (1, 4, 8 and 24-bit) with compression (RLE, Group 3 fax, and Group 4 fax, Pack Bits, LZW, JPEG).

Note: Not all actions that manipulate images support the same input file formats as this action.

Example:

```
SetLuminanceFactor ("24")  
SetChrominanceFactor ("10")  
SetGrayScale ("True")  
ConvertToJPEG ()
```

Parent topic: [ImageConvert actions](#)

ConvertToTIFF

Converts the current image to a TIFF.

Syntax

```
bool ConvertToTIFF ()
```

Parameters

None.

Returns

True, if the image is converted to a TIFF. False, if the input image is a TIFF or if a failure occurs during the conversion.

Level

Page Level.

Details

Converts the current image to a TIFF. The output file name will have a TIF extension.

Supported input file formats: BMP (1, 4, 8, or 24-bit), GIF (1, 4 or 8-bit), JPG or JPEG (8 and 12-bit grayscale and 24-bit color), and PNG (1, 4, 8 and 24-bit).

Note: Not all actions that manipulate images support the same input file formats as this action.

Example:

```
SetTiffCompression ("3")  
ConvertToTIFF ()
```

Parent topic: [ImageConvert actions](#)

RescaleImage

Adjusts image for the current page to the specified size.

Syntax

```
bool RescaleImage (string ImageHeight, string ImageWidth, string ImageDPI)
```

Parameters

Smart parameters are supported.

ImageHeight

Output Height of the page in pixels. If blank, defaults to 3300 pixels.

ImageWidth

Output Width of the page in pixels. If blank, defaults to 2550 pixels.

ImageDPI

The new DPI of the image. If blank, defaults to 300 dpi.

Returns

True, if the operation is successful, else False, if any error occurs.

Level

Page level only, and the page must refer to a valid single page image file.

Details

Scales the image and sets the DPI to the specified values.

Example

```
RescaleImage (3300, 2550, 300)
```

Parent topic: [ImageConvert actions](#)

SetChrominanceFactor

Set Compression for JPEG Output Files.

Syntax

```
bool SetChrominanceFactor (StrParam)
```

Parameters

A value of 0 to 255, with a value of 0 meaning minimum compression (best quality) and 255 meaning maximum image compression. Smart Parameters are supported.

Returns

Always True.

Level

All levels.

Details

JPEG images are stored in a compressed format. This action controls the amount of compression that is used when you are converting an image to a JPEG image using the `ConvertToJPEG` action.

Data loss is the result of JPEGs ability to achieve high compression ratios. Higher-quality settings result in less compression, while lower quality settings result in higher compression. Quality versus compression is a trade-off. Adjust image compression ratio by setting the `SetLuminanceFactor` and `SetChrominanceFactor` actions. The `SetLuminanceFactor` action adjusts the luminance or gray scale quality, while the `SetChrominanceFactor` action adjusts the chrominance or color quality. Lower settings for these properties result in higher-quality images with less compression. Higher settings for these properties result in lower quality images with more compression.

If this action is not called, a default value of 10 is used.

Example:

```
SetLuminanceFactor ("24")
SetChrominanceFactor ("10")
SetGrayScale ("True")
ConvertToJPEG ()
```

Parent topic: [ImageConvert actions](#)

Related reference:

[ConvertToJPEG](#)

SetDeleteOriginal

Controls deletion of file after conversion.

Syntax

```
bool SetDeleteOriginal (StrParam)
```

Parameters

True: After the new image is created, the original will be deleted. False: After the new image is created, the original will remain.

Smart Parameters are supported.

Returns

Always True.

Level

All levels.

Details

Use this action to control the deletion of the source image file when using the actions `ConvertToJPEG` and `ConvertToTIFF`. If this action is not called, the default value of False will be used, the source image will not be deleted after conversion.

Example:

```
SetDeleteOriginal ("True")
ConvertToJPEG ()
```

Parent topic: [ImageConvert actions](#)

Related reference:

[ConvertToJPEG](#)

[ConvertToTIFF](#)

SetGrayScale

Controls the grayscale output for JPEG images.

Syntax

```
bool SetGrayScale (StrParam)
```

Parameters

True: Saves the image as a grayscale image. False: Saves the image as color.

Smart Parameters are supported.

Returns

Always True.

Level

All Levels.

Details

This action will determine if a JPEG image will be output as grayscale or color. If SetGrayScale is True, all images will be saved as 8 bit grayscale. If SetGrayScale is False, color images will be saved as 24 bit color and if the original image is 1 bit black and white, the new image will be 8 bit grayscale. If this action is not called, the value will default to False (color).

Example:

```
SetLuminanceFactor ("24")
SetChrominanceFactor ("10")
SetGrayScale ("True")
ConvertToJPEG ()
```

Parent topic: [ImageConvert actions](#)

Related reference:

[ConvertToJPEG](#)

SetImageDPIByWidth

Adjusts the DPI of the image and resizes the image based on the expected physical size.

Syntax

```
bool SetImageDPIByWidth (string PageWidth, string ImageDPI, string Rescale)
```

Parameters

Smart parameters are supported.

PageWidth

Physical width of the page in inches. If not specified, the value defaults to 8.5 inches.

ImageDPI

The target DPI for the updated image. If not specified, the value defaults to 300 dpi.

Rescale

Indicates whether the image must be rescaled or not.

- True - Rescale image width and height to match the specified DPI and physical size.
- False - Do not rescale the page, set the DPI only.

Returns

True, if the operation is successful, else False, if any error occurs.

Level

Page level only, and the page must refer to a valid single page image file.

Details

This action adjusts the DPI and width of the image to the specified DPI. The image can also be resized to the correct logical size based on the expected physical width in inches. This action is useful for images that do not have a correctly specified DPI or needs to be resized based on the expected width.

When the rescale parameter is true, the image size is adjusted. The image is scaled proportionally in the horizontal and vertical directions based on the width that is specified for the image. The final image has the specified DPI and the pixel width and height is scaled to the specified logical size.

Example

```
SetImageDPIByWidth ("6", "", "True")
```

Parent topic: [ImageConvert actions](#)

SetLuminanceFactor

Sets the image luminance or grayscale quality.

Syntax

```
bool SetLuminanceFactor (StrParam)
```

Parameters

A value of 0 to 255, with a value of 0 meaning minimum compression (best quality) and 255 meaning maximum image compression. Smart Parameters are supported.

Returns

Always True.

Level

All levels.

Details

JPEG images are stored in a compressed format. This action controls the amount of compression that is used when you are converting an image to a JPEG image using the `ConvertToJPEG` action.

Data loss is the result of JPEGs ability to achieve high compression ratios. Higher-quality settings result in less compression, while lower quality settings result in higher compression. Quality versus compression is a trade-off. Adjust image compression ratio by setting the `SetLuminanceFactor` and `SetChrominanceFactor` actions. The `SetLuminanceFactor` action adjusts the luminance or gray scale quality, while the `SetChrominanceFactor` action adjusts the chrominance or color quality. Lower settings for these properties result in higher-quality images with less compression. Higher settings for these properties result in lower quality images with more compression.

If this action is not called, a default value of 24 is used.

Example:

```
SetLuminanceFactor ("24")
SetChrominanceFactor ("10")
SetGrayScale ("True")
ConvertToJPEG ()
```

Parent topic: [ImageConvert actions](#)

Related reference:

[ConvertToJPEG](#)

SetTIFFCompression

Controls the compression format when saving a TIFF.

Syntax

```
bool SetTIFFCompression (StrParam)
```

Parameters

One of the following values:

- 0 : No compression.
- 1 : Run length encoding (RLE).
- 2 : CCITT Group 3 fax compression.
- 3 : CCITT Group 4 fax compression.
- 4 : LZW compression.
- 5 : Apple Macintosh PackBits compression.
- 6 : JPEG compression format.
- 7 : Lossless compression standard derived from LZ77.

- 8 : CCITT Group 3 two-dimensional standard fax compression.

Smart Parameters are supported.

Returns

Always True.

Level

All levels.

Details

This action sets the compression type of the TIFF that is output from the action `ConvertToTIFF`. Typically Group 4 compression is used for performing recognition on images.

All of these possible output formats may not be supported by other image processing actions. If this action is not called, Group 4 compression will be used.

Example:

```
SetTiffCompression("3")
ConvertToTIFF()
```

Parent topic: [ImageConvert actions](#)

Related reference:

[ConvertToTIFF](#)

ImageFix actions

The ImageFix actions are older versions of the DCImageFix action. Use the DCImageFix actions instead.

Parent topic: [Global actions](#)

Imail actions

Use the Imail actions to import image attachments from a mail server into the current batch by using IMAP.

Important: If any concurrently running threads might be using the same mail account, your Imail-related tasks cannot be run in a multi-threaded configuration. Instead, use single-threaded tasks only. For information about configuring threads in Rulerunner, see [Rulerunner thread configuration](#). For information about the mail account, see [im_login](#).

The Imail scan action polls the server and imports image attachments until the batch reaches the specified size or until the wait time expires.

- [im_abort_time](#)
Specifies the delay time before it returns when a batch stops running.
- [im_AcceptMixedAttachments](#)
Ingest emails with selected attachment types, even if non-selected attachment types are included
- [im_AcceptNoAttachments](#)
Ingest emails without attachments, even if attachment types are specified

- [im_done_folder](#)
Specifies the IMAP folder for successfully imported emails.
- [im_login](#)
Specifies the IMAP mail server URL and login credentials.
- [im_logout](#)
Disconnect from the mail server.
- [im_max_docs](#)
Specifies maximum number of emails to include in a single batch.
- [im_problem_folder](#)
Specifies the IMAP folder for problem emails.
- [im_reject_types](#)
Specifies file extensions for attachments to reject.
- [im_scan](#)
Poll the specified mail server for incoming emails with image attachments.
- [im_SetProxy](#)
Specifies the settings for a proxy service that can be used to connect to the email server.
- [im_SortByDate](#)
Sort emails by received date
- [im_StoreEML](#)
Store one .eml file per message, rather than one file per attachment
- [im_types](#)
Specifies valid file extensions for image attachments.
- [im_UseSSL](#)
Connect to IMAP Server by using an SSL encrypted channel
- [im_wait_time](#)
Specifies the maximum time to wait for input emails for a single batch.

Parent topic: [Global actions](#)

im_abort_time

Specifies the delay time before it returns when a batch stops running.

Member of namespace

Imail

Syntax

```
bool im_abort_time(int nSecs)
```

Parameters

nSecs
Type: int

Parameters

nSecs: The number of seconds to wait.

Returns

Always True.

Level

Batch level.

Details

The action waits the specified time before it returns when an abort occurs. This action can be useful to prevent many aborted batches due to an abort condition. For example, if the email server can become unavailable for a time, the abort timeout limits the number of aborted batches until the mail server becomes available again.

If this action is not called, the default abort time value of 30 seconds is used.

Example:

```
im_wait_time("20")
im_abort_time("60")
im_max_docs("200")
im_types("jpg,tif")
```

Parent topic: [Imail actions](#)

im_AcceptMixedAttachments

Ingest emails with selected attachment types, even if non-selected attachment types are included

Member of namespace

Imail

Syntax

```
bool im_AcceptMixedAttachments(bool bMixedOK)
```

Parameters

bMixedOK

Type: bool

A boolean value that enables ingestion of emails with both allowed and disallowed attachment types.

Parameters

True An email message with an attachment type not specified by `im_types` is ingested when it contains a required attachment.

False An email message with any disallowed attachment is rejected, which is the default process if this action is never called.

Returns

Always True.

Level

Batch level.

Details

If this action is never called, the presence of any attachment that is not specified by `im_types` causes `im_scan` to treat that email message as a Problem.

If this action is called with parameter `True`, messages with disallowed attachment types are ingested when they contain at least one allowed attachment.

If `im_types` is called with a blank parameter before `im_scan`, all emails are ingested, so this action has no effect.

Parent topic: [Iemail actions](#)

im_AcceptNoAttachments

Ingest emails without attachments, even if attachment types are specified

Member of namespace

Iemail

Syntax

```
bool im_AcceptNoAttachments (bool bNoAttachOK)
```

Parameters

`bNoAttachOK`

Type: bool

A boolean value that enables ingestion of emails with no attachments.

Parameters

True: An email message with no attachments is ingested by `im_scan`.

False: An email message no attachments are rejected by `im_scan`, unless `im_types` was called with a blank parameter, which is the default behavior if this action is never called.

Returns

Always True.

Level

Batch level.

Details

If this action is never called, the absence of an attachment causes `im_scan` to treat that email message as a Problem.

If this action is called with parameter True, a message with no attachments are ingested.

If `im_types` is called with a blank parameter before `im_scan`, all emails are ingested, so this action has no effect.

Parent topic: [Iml actions](#)

im_done_folder

Specifies the IMAP folder for successfully imported emails.

Member of namespace

Iml

Syntax

```
bool im_done_folder(string folder)
```

Parameters

folder
Type: string

Parameters

folder: Destination IMAP folder for successfully imported emails.

Returns

Always True.

Level

Batch level.

Details

When an email is processed and the attachment is imported, the email is moved to the folder name specified by this action. The folder is expected to be on the root level, the same level as the inbox folder in the mail account that is specified by the `im_login` action. If the folder exists in a subfolder, then use a forward slash to separate the folder names.

If this action is not called, the default value of Done is used.

Example:

```
im_done_folder("Imported")  
im_problem_folder("Failed")
```

This example changes the default names of the Done and Problem folders and also requires the Imported and Failed folders to be on the same level as the Inbox folder.

```
im_done_folder("Inbox/Imported")
im_problem_folder("Inbox/Failed")
```

This example requires the folders to be a subfolder of the Inbox folder.

Parent topic: [Email actions](#)

im_login

Specifies the IMAP mail server URL and login credentials.

Member of namespace

Email

Syntax

```
bool im_login(string hostname, string username, string password)
```

Parameters

Smart parameters are supported for all parameters.

hostname

A string that specifies the following attributes of the IMAP mail server:

String inclusion	Attribute
Required	The URL or IP address.
Optional	The port number. This number is at the end of the string and preceded by a colon.

username

The username for the mail account.

password

The password for the mail account.

Returns

True If the login succeeds. Otherwise, False.

Level

Batch level Open event only.

Details

Connects to the mail server by using the specified account information. Specifies the mail server URL and login credentials. Login credentials are for a mail server that supports the IMAP protocol (such as MS Exchange, and many others). Call one time for each batch.

The password can be encrypted by storing it in the Advanced section of the Custom tab in the Datacap Application Manager. The URL and user name parameters can be stored in the General section of the Application Manager but they cannot be encrypted. To retrieve the password that is stored in the Advanced section, use a smart parameter for the action's password parameter similar to

@APPVAR(values/adv/mailPassword) where mailPassword is the name that is entered for the value in the Application Manager.

You can affect the behavior of this action by first calling the following actions:

To connect on an encrypted channel by using the SSL / HTTPS protocol	Call im_useSSL(True).
To connect by using a proxy service	Call im_SetProxy.

Example:

```
im_login("mymailserver.com","theuser","password")
im_login("mymailserver.com:9080","theuser","password")
```

Parent topic: [Iemail actions](#)

Related reference:

[im_SetProxy](#)

[im_UseSSL](#)

im_logout

Disconnect from the mail server.

Member of namespace

Iemail

Syntax

```
bool im_logout ()
```

Parameters

None.

Returns

Always True.

Level

Batch level Open or Close event only.

Details

Closes the connection to the mail server. Call one time for each batch after im_login and im_scan are completed.

Example:

```
im_logout ()
```

Parent topic: [Iemail actions](#)

im_max_docs

Specifies maximum number of emails to include in a single batch.

Member of namespace

Imail

Syntax

```
bool im_max_docs(int nDocs)
```

Parameters

nDocs
Type: int

Parameters

nDocs: The maximum number of emails in a batch.

Returns

Always True.

Level

Batch level.

Details

The import of emails into the batch stops when this email limit is reached or when the maximum wait time is reached. While it is waiting for new mail to arrive, the configured mailbox is polled every 2 seconds to check for waiting mail.

If this action is not called, the default value of 100 is used. The actual amount included in the batch might be less than this maximum.

Example:

```
im_wait_time("20")
im_abort_time("60")
im_max_docs("50")
im_types("jpg,tif")
```

This example causes the scan operation to limit the number of emails in a batch to a maximum of 50.

Parent topic: [Imail actions](#)

im_problem_folder

Specifies the IMAP folder for problem emails.

Member of namespace

Iemail

Syntax

```
bool im_problem_folder(string folder)
```

Parameters

folder
Type: string

Parameters

folder: Destination IMAP folder for problem emails.

Returns

Always True.

Level

Batch level.

Details

When an email is processed and the attachment is not one of the expected types, the email is moved to the folder specified by this action.

The folder is expected to be on the root level, the same level as the inbox folder of the mail account that is specified by the `im_login` action. If the folder exists in a subfolder, then use a forward slash to separate the folder names.

If this action is not called, the default value of Problem is used.

Example:

```
im_done_folder("Imported")
im_problem_folder("Failed")
```

This example changes the default name of the Problem folder to Failed and requires the Imported and Failed folders to be on the same level as the Inbox folder.

```
im_done_folder("Inbox/Imported")
im_problem_folder("Inbox/Failed")
```

This example specifies the folder, which is a subfolder of the Inbox folder.

Parent topic: [Iemail actions](#)

im_reject_types

Specifies file extensions for attachments to reject.

Member of namespace

Email

Syntax

```
bool im_reject_types (string extensions)
```

Parameters

extensions - Comma-separated list of file image file extensions to ignore, with or without period. Use in conjunction with `im_AcceptMixedAttachments(True)` to reject emails with specific attachment types. Attachments specified in `im_types` are ingested, attachment types not specified in `im_types` or `im_reject_types` are ignored. Smart parameters are supported.

Returns

Always True.

Level

Batch level.

Details

This action specifies disallowed attachment types.

Example:

```
im_types("jpg,tif")
im_AcceptMixedAttachments("True")
im_reject_types(".eml")
```

Parent topic: [Email actions](#)

im_scan

Poll the specified mail server for incoming emails with image attachments.

Member of namespace

Email

Syntax

```
bool im_scan ()
```

Parameters

None.

Returns

False If the operation fails, and pauses before it returns. Otherwise, True.

If no selected emails were available, the action returns True and pauses before it returns.

Action returns when timeout is reached, or the requested number of emails are processed.

Level

Batch level Open event only.

Details

Scans emails in the Inbox for specified attachments, imports selected emails with attachments into the batch. Call one time for each batch. A connection to the email server must be previously established by using the `im_login` action.

Each input email contains the following variables set in the document hierarchy:

- **TYPE:** Always set to "Document".
- **Subject:** The email subject.
- **Body:** The text within the email.
- **DateSent:** The sent date stamp on the email.
- **From:** The email sender.
- **User:** The email user who is specified in the `im_login` action.
- **To:** The email recipients.
- **Priority:** The state of the email importance flag.

Attachments are ingested as individual pages within the email document, unless `im_StoreEML(true)` is called before `im_scan`. With individual attachments, each attachment page has the following variables set.

- **TYPE:** Always set to "Other".
- **IMAGEFILE:** The name of the attachment as saved on disk.

The following batch level variable is created:

- **EmailCount:** The number of emails that are scanned into the batch.

It is recommended that the subject line is no longer than 78 characters because this length is a common subject line length limitation. Some systems might support even shorter lengths, truncating the subject. Testing was successful with lengths up to 255 characters. It is recommended to test your settings and use lengths appropriate for your systems.

Example:

```
im_wait_time("20")
im_abort_time("40")
im_max_docs("200")
im_types("jpg,tif")
im_scan()
```

Parent topic: [Iml actions](#)

Related reference:

[im_abort_time](#)

[im_AcceptMixedAttachments](#)

[im_AcceptNoAttachments](#)

[im_done_folder](#)

[im_problem_folder](#)

[im_SortByDate](#)

im_StoreEML
im_UseSSL
im_wait_time

im_SetProxy

Specifies the settings for a proxy service that can be used to connect to the email server.

Member of namespace

Imail

Syntax

```
bool im_SetProxy(string hostname, string proxyPort, string proxyType, string proxyUsername, string proxyPassword)
```

Parameters

Smart parameters are supported for all parameters.

hostname

The URL or IP address of the proxy service.

You can store the passed parameter value in the General section of the Datacap Application Manager.

proxyPort

The proxy service connection port.

proxyType

The type of proxy service. The following values are possible:

- HTTP
- SOCKS4
- SOCKS5
- NONE

Any unsupported type is treated as NONE.

proxyUsername

The username for the proxy server.

You can store the passed parameter value in the General section of the Application Manager.

proxyPassword

The password for the proxy server.

You can store the passed parameter value in the Advanced section of the Custom tab in the Application Manager. A smart parameter such as @APPVAR(values/adv/<mailProxyPassword>) retrieves the value, where <mailProxyPassword> is the name that you entered for the value in the Application Manager.

Returns

False if the proxyPort parameter value is not numeric. Otherwise, the action returns True.

Important: This action does not attempt to log on to the proxy service. The value True is returned regardless of the validity of the passed proxy service settings. If the settings are invalid, this condition is detected when you

call `im_login`.

Level

Any level but typically batch level.

Details

Use this action to optionally specify a proxy service that connects to the email server. The `im_login` action uses the specified proxy service settings to connect to the proxy service. If you do not call this action before the call to `im_login`, no proxy service is used.

Example:

```
im_SetProxy("theHostName", "1234", "SOCKS4",  
"MyUser", "@APPVAR(values/adv/mailProxyPassword)")  
im_login("mymailserver.com", "theuser", "password")
```

In this example, the call to `im_SetProxy` specifies the settings for a proxy service, which the `im_login` action then uses to connect to that service. As illustrated, the smart parameter `APPVAR` can be used to pass in the `proxyPassword` parameter value for `im_SetProxy`.

Parent topic: [Imail actions](#)

Related information:

[Smart parameters](#)

im_SortByDate

Sort emails by received date

Member of namespace

Imail

Syntax

```
bool im_SortByDate (bool bSort)
```

Parameters

`bSort`

Type: `bool`

A boolean value that enables or disables sorting of emails before ingestion.

Parameters

True: Causes `im_scan` to sort email messages by date they are received and ingested the oldest messages first. This sorting is the default behavior if `im_SortByDate` is never called.

False: Disables sorting of email messages. This sorting greatly increases performance of `im_scan` when many emails are pending in the inbox.

Returns

Always True.

Level

Batch level.

Details

When you enable sorting, all pending emails are sorted and the oldest is ingested first. This process can take a significant amount of time for each batch. When the inbox might contain many emails (for example, more than 100), call `im_SortByDate(False)` to speed up the process. In this case, messages are processed in the order that the email server presents them to Datacap.

Parent topic: [Iemail actions](#)

im_StoreEML

Store one .eml file per message, rather than one file per attachment

Member of namespace

Iemail

Syntax

```
bool im_StoreEML(bool bStoreEML)
```

Parameters

bStoreEML

Type: bool

A boolean value that enables storing each email message as an .eml file.

Parameters

True: Store each ingested email message, formatted together with all attachments, in a single .eml file and attach it as a page to the document.

False: Add each ingested email message as a document, with associated variables, and add each attachment as a page within the document. This process is the default if `im_StoreEML` is not called.

Returns

Always True.

Level

Batch level.

Details

If set, the `im_scan` action creates a one page document that contains the email and attachments in an .eml file. No attachment pages are created and no variables are set in the document.

im_types

Specifies valid file extensions for image attachments.

Member of namespace

Iemail

Syntax

```
bool im_types(string extensions)
```

Parameters

extensions
Type: string

Parameters

extensions: Comma-separated list of file image file extensions to import, with or without period. If the parameter is blank, all emails with or without attachments are processed.

Returns

Always True.

Level

Batch level.

Details

This action specifies the allowable email attachment types.

If this action is not called, the default value of .pdf is used.

If this action is called and no attachment types are specified, all emails and any attachments are added to the batch. Emails that are processed without attachments result in documents without any pages.

If attachment types are specified and the email contains:

- Only the specified attachment types, the email, and attachments are added to the batch
- Only unspecified attachment types, or if it contains a mix of specified and unspecified attachment types, the email is moved to the Problem folder
- No attachments, the email is moved to the Problem folder

Example:

```
im_wait_time("20")
im_abort_time("40")
im_max_docs("200")
im_types("jpg,tif")
```

Parent topic: [Iemail actions](#)

im_UseSSL

Connect to IMAP Server by using an SSL encrypted channel

Member of namespace

Iemail

Syntax

```
bool im_UseSSL(bool bUseSSL)
```

Parameters

bUseSSL

Type: bool

A boolean value that enables or disables SSL communication.

Parameters

True: Use SSL to encrypt communication with the mail server.

False: Communicate with the email server without encryption, which is the default method if im_UseSSL is not called.

Returns

Always True.

Level

Batch level.

Details

When SSL is enabled, subsequent im_login and im_scan actions communicate with the email server over IMAPS protocol by using SSL encryption on port 993. When SSL is not in use, these actions communicate with the email server over IMAP protocol on port 143.

Parent topic: [Iemail actions](#)

im_wait_time

Specifies the maximum time to wait for input emails for a single batch.

Member of namespace

Iemail

Syntax

```
bool im_wait_time(int nSecs)
```

Parameters

nSecs
Type: int

Parameters

nSecs: The maximum number of seconds to wait.

Returns

Always True.

Level

Batch level.

Details

The maximum time to wait for input emails for a single batch, when there is at least one pending email. Used by the `im_scan` action after the first email is processed to determine how long to wait for the batch to fill up. If no emails are pending, the `im_scan` action does not wait regardless of the wait time value.

The import of emails into the batch stops when the wait limit is reached or when the maximum email per batch is reached. While it is waiting for new mail to arrive, the configured mailbox is polled every 2 seconds to check for waiting mail. The action continues to include new files into the batch until this wait time is reached or the maximum number of emails per batch is reached.

If this action is not called, the default wait time of 5 seconds is used.

Example:

```
im_wait_time("20")
im_abort_time("60")
im_max_docs("200")
im_types("jpg,tif")
```

Parent topic: [Email actions](#)

Imprint actions

Use the Imprint actions to imprint text over an image, or for blackout or whiteout redactions.

The Imprint actions can specify the width, font style, and font size of the imprinted text as well as whether overlay rectangles are opaque or transparent.

- [AnnotateImage](#)
Imprints the text that you specify onto the current page image.
- [RedactByRegEx](#)
Searches the entire page and redacts all occurrences of the phrase that matches the RegEx expression.
- [RedactParameters](#)
Overlays a black or white rectangle on the image to redact the current field or the specified region of a page.

- [SetAdjustedWidth](#)
Specifies the width of the imprinted text.
- [SetFontName](#)
Specifies the font style to use for the imprinted text.
- [SetFontSize](#)
Specifies the font size to use for the imprinted text.
- [SetOpaque](#)
Specifies whether the transparency of the overlay rectangles are opaque (1) or transparent (0).

Parent topic: [Global actions](#)

AnnotateImage

Imprints the text that you specify onto the current page image.

Syntax

```
bool AnnotateImage (string displayText, string xCoordinate, string yCoordinate)
```

Parameters

string displayText

string xCoordinate

string yCoordinate

Parameters

Three parameters:

- displayText smart parameter supported string to be placed onto the image. The displayText can combine plain text along with smart parameters.
- xCoordinate The X coordinate for the starting position of the text on the image. Smart parameter supported.
- yCoordinate The Y coordinate for the starting position of the text on the image. Smart parameter supported.

Returns

False, if parameters are missing, or the X or Y parameters are not Numeric. Otherwise, True.

Level

Page level only.

Details

Imprints the text you specify onto the current page's image file. By default, the text's font size is 12, and font's style Times New Roman, with an adjusted width of 30.

Example:

```
SetFontName("Arial")
SetFontSize("10")
SetAdjustedWidth("100")
AnnotateImage("@BATCHID+ Page:+@ID", "0", "0")
```

This example places the Batch ID followed by `Page:` and the calling object ID at the top of the image using the Arial font with a point size of 10 and a width of 100.

Parent topic: [Imprint actions](#)

RedactByRegEx

Searches the entire page and redacts all occurrences of the phrase that matches the RegEx expression.

Syntax

```
bool RedactByRegEx (string RegEx, string VariableBase)
```

Parameters

string RegEx

string VariableBase

Parameters

- RegEx The expression to search for on the page. Smart parameters are supported.
- VariableBase A variable to create in the runtime DCO to track each redaction.

Returns

Always True.

Level

Page Level.

Details

This action searches the entire page and redacts all occurrences of the phrase that matches the RegEx expression. This action must be called at a field level, however the coordinates of the field will change during the operation and will remain on the last found occurrence of the found text, so it may be necessary to use a zoned but unused field for this action.

The VariableBase parameter specifies a variable name to use as a history of redacted words. For each redaction, a variable of the form "VariableBasen", where n is an incrementing number, is created in the runtime DCO with the coordinates of the redacted word. For example, if the parameter was "Hello", the variables created would be named: "Hello1", "Hello2", "Hello3", etc. for each redaction. An example value of the variable: "TM000001,159,1085,242,1115".

Example:

```
RedactByRegEx("[Hh]istory", "Location")
```

This example will redact all areas on the page where the word "History" or "history" exists. The redactions will be tracked in the runtime DCO in a variable called Location0, Location1, etc.

Parent topic: [Imprint actions](#)

Related reference:
[Redact \(deprecated\)](#)

RedactParameters

Overlays a black or white rectangle on the image to redact the current field or the specified region of a page.

Syntax

```
bool RedactParameters (string FillColor, string NewText, string TopLeftX, string TopLeftY, string BottomRightX, string BottomRightY)
```

Parameters

string

string FillColor

string NewText

string TopLeftX

string TopLeftY

string BottomRightX

string BottomRightY

Parameters

- FillColor The fill color to use. Must be either White, or Black.
- NewText Optional - The text to include in the overlay.
- TopLeftX The upper left X coordinate in pixels.
- TopLeftY The upper left Y coordinate in pixels.
- BottomRightX The lower right X coordinate in pixels.
- BottomRightY The lower right Y coordinate in pixels.

Smart parameters are supported for all parameters.

Returns

True, if the area is redacted. Otherwise, False.

Level

Page and Field level.

Details

This action overlays a black, or white rectangle on the image. A default text value might optionally be applied to the overlay.

If run on the field level, the entire field is redacted and the last four X/Y parameters are ignored.

If run on the page level the last four parameters to specify the location must be provided.

Example:

```
Redact("black", "", "0", "0", "100", "100")
```

Called at the page level, this applies a black square overlay on the upper left of the image.

```
Redact("White", "Hello", "", "", "", "")
```

Called at the field level, this applies a white square overlay on the coordinates for the current field, and imprint the word "Hello".

Parent topic: [Imprint actions](#)

SetAdjustedWidth

Specifies the width of the imprinted text.

Syntax

```
bool SetAdjustedWidth (StrParam)
```

Parameters

Numeric value for a length adjustment of the string to be imprinted. Smart parameters are supported.

Returns

False if the parameter is not numeric. Otherwise, True.

Level

Page level only.

Details

This actions adjusts the maximum width of the imprinted text. It is possible that the calculation of the string length may not allow for the entire string in the specified font and point size. This is an adjustment factor which will lengthen the area for the string to imprint on the image.

This action is optional. If the action is not used, the adjusted width will default to 30. If your text is being cut off, increase the parameter value. If you use this action it must be called prior to the AnnotateImage action.

Example:

```
SetFontName("Arial")
SetFontSize("10")
SetAdjustedWidth("100")
AnnotateImage("@BATCHID+ Page:+@ID", "0", "0")
```

Parent topic: [Imprint actions](#)

SetFontName

Specifies the font style to use for the imprinted text.

Syntax

```
bool SetFontName (StrParam)
```

Parameters

String value of the font's name. Smart parameters are supported.

Returns

False if the parameter is missing. Otherwise True.

Level

Page level only.

Details

Specifies the font style that is used.

This action is optional. If not used, the font style will default to Times New Roman. If you use this action, it must be called prior to the AnnotateImage action.

Example:

```
SetFontName("Arial")
SetFontSize("10")
SetAdjustedWidth("100")
AnnotateImage("@BATCHID+ Page:+@ID", "0", "0")
```

Parent topic: [Imprint actions](#)

SetFontSize

Specifies the font size to use for the imprinted text.

Syntax

```
bool SetFontSize (StrParam)
```

Parameters

Numeric value of the font's size. Smart parameters are supported.

Returns

False if the parameter is not Numeric. Otherwise True.

Level

Page level only.

Details

Specifies the font size that will be used.

This action is optional. If you do not use the action, the font size will default to 12. If you use this action, it must be called prior to the AnnotateImage action.

Example:

```
SetFontName("Arial")
SetFontSize("10")
SetAdjustedWidth("100")
AnnotateImage("@BATCHID+ Page:@ID", "0", "0")
```

Parent topic: [Imprint actions](#)

SetOpaque

Specifies whether the transparency of the overlay rectangles are opaque (1) or transparent (0).

Syntax

```
bool SetOpaque (StrParam)
```

Parameters

- Integer value: 1 or 0.
- SetOpaque(1) indicates full opacity / black.
- SetOpaque(0) results in subsequent rectangles that are white.

Smart parameters are supported.

Returns

False if the parameter is not an Integer. Otherwise, True.

Level

All, but generally used at the Page level.

Details

Sets the transparency of the background of the imprint text. This is the background that is added behind the imprinted text. If you use this action it must be called prior to the AnnotateImage action.

If this action is not called, then the default value of 0, white, will be used for the background of the annotation text.

Example:

```
SetOpaque ("1")
AnnotateImage ("@BATCHID+ Page:+@ID", "0", "0")
```

Parent topic: [Imprint actions](#)

Intellocate actions

Use the Intellocate actions to update the existing field position information in the document hierarchy (setup DCO) or to add position information for a new fingerprint.

The Intellocate actions can generate fingerprints automatically by using a page image and the recognition zones that are specified manually during verification.

- [iloc_AdjustZones](#)
Updates fingerprint-specific position coordinates for Field objects in the Document Hierarchy based on the locations listed for the current source page's Data file (.xml).
- [iloc_AssignPageType](#)
Assigns a required Page Type value to a newly created fingerprint.
- [iloc_SetDetailZones](#)
Writes the position coordinates of a new fingerprint's Detail Line fields from a page's Data file to the Pos properties of the corresponding Detail Line Field objects of the application's Document Hierarchy.
- [iloc_SetZones](#)
Writes the position coordinates of a new fingerprint's zoned fields from a page's Data file to the Pos properties of the corresponding Field objects in the Document Hierarchy file (.xml).
- [IsPageDataMissing](#)
Checks to see that the current page data exists.

Parent topic: [Global actions](#)

iloc_AdjustZones

Updates fingerprint-specific position coordinates for Field objects in the Document Hierarchy based on the locations listed for the current source page's Data file (.xml).

Member of namespace

Intellocate

Syntax

```
bool iloc_AdjustZones ()
```

Parameters

None.

Returns

False if a fingerprint match has not occurred and a Template ID value has not been assigned to the current page, or if the Document Hierarchy file cannot be saved. Otherwise, True.

Level

Page level only.

Details

Updates fingerprint-specific position coordinates for Field objects in the Document Hierarchy based on the locations listed for the current source page's Data file (.xml).

This action is designed to be used on existing fingerprints only. It will only update fields that do not have position information. Existing fields will not be updated, even if the position information for those fields has changed.

Example:

```
iloc_AdjustZones()
```

Parent topic: [Intellocate actions](#)

iloc_AssignPageType

Assigns a required Page Type value to a newly created fingerprint.

Member of namespace

Intellocate

Syntax

```
bool iloc_AssignPageType (strParam)
```

Parameters

The value of the page type to assign to the current page. The input parameter must be one of the following:

1. A numeric value that corresponds to the Page Type.
2. A string value of the page type name.

Returns

False if the numeric value cannot be retrieved from the fingerprint database or if there is no connection to the fingerprint database. Otherwise, True.

Level

Page level only.

Details

Assigns a required Page Type value to a newly created fingerprint. The assigned value corresponds to the values configured in the PageType table within the fingerprint database. These values are custom defined for each application. If a string value is passed as a parameter, this action will look up the corresponding numeric value within the database.

Example:

```
Iloc_AssignPageType ("2")
```

This example assigns a page type of 2 to the new fingerprint.

```
Iloc_AssignPageType ("PageSeparator")
```

This example assigns a page type of "PageSeparator" to the new fingerprint.

Parent topic: [Intellocate actions](#)

iloc_SetDetailZones

Writes the position coordinates of a new fingerprint's Detail Line fields from a page's Data file to the Pos properties of the corresponding Detail Line Field objects of the application's Document Hierarchy.

Member of namespace

Intellocate

Syntax

```
bool iloc_SetDetailZones ()
```

Parameters

None.

Returns

Always True.

Level

Page level only.

Details

Writes the position coordinates of a new fingerprint's Detail Line fields from a page's Data file to the Pos properties of the corresponding Detail Line Field objects of the application's Document Hierarchy.

Example:

```
iloc_SetDetailZones ()
```

Parent topic: [Intellocate actions](#)

iloc_SetZones

Writes the position coordinates of a new fingerprint's zoned fields from a page's Data file to the Pos properties of the corresponding Field objects in the Document Hierarchy file (.xml).

Member of namespace

Intellocate

Syntax

```
bool iloc_SetZones ()
```

Parameters

None.

Returns

False if a fingerprint match has not occurred, or if the Document Hierarchy file (.xml) cannot be saved. Otherwise, True.

Level

Page level only.

Details

Writes the position coordinates of a new fingerprint's zoned fields from a page's Data file to the Pos properties of the corresponding Field objects in the Document Hierarchy file (.xml).

Example:

```
iloc_SetZones ()
```

Parent topic: [Intellocate actions](#)

IsPageDataMissing

Checks to see that the current page data exists.

Member of namespace

Intellocate

Syntax

```
bool IsPageDataMissing ()
```

Parameters:

None.

Returns

True if the current page does not have page data. Otherwise, False.

Level

Page level only.

Details

Checks to see that the current page data exists. This action does not physically check for the existence of a data file. It confirms that a valid page data is loaded into memory.

Example:

```
IsPageDataMissing()
```

Parent topic: [Intellocate actions](#)

Invoice actions

Use the Invoice actions to process invoices by using the Datacap Accounts Payable application.

The Invoice actions can configure date, fingerprint, format, and vendor settings on invoice pages before export.

- [AddToDetailErrorMsg](#)
Appends extra text to the existing error message.
- [AddToErrorMsg](#)
Appends the supplied text to any existing error message string.
- [AllMixedCase](#)
Changes the value of the current field to be Title case. The first letter of each word is capitalized.
- [AlterDatebyDay](#)
Adds the specified number of days to the date contained in the current field. The original character confidence is not changed.
- [CalculateNotesZone](#)
Creates a zone between detail lines to recognize the text between lines.
- [CaptureOpInfo](#)
Captures the Operator, Station ID, and current time and place them into variables. The variables are named based on the provided variable prefix, such as, *prefix Operator*, *prefix Station*, and *prefix Time*. When, a prefix is not provided the task name is used as the prefix by default.
- [CheckAndFixDecimal](#)
Replaces the space or comma with a decimal point. This action is used to fix errors where the decimal is not recognized and leaves a blank in that area.
- [CheckForSticky](#)
For new fingerprints, checks whether there is another matching fingerprint within the same batch that was already verified. It can be used to obtain zone information.
- [CheckFreeDiskSpace](#)
Reads the LowDiskSpaceThreshold setting from the [Notifications] section of the INI file that is specified in the first parameter.
- [ClearErrorMsg](#)
Use this action to clear the current error message.
- [CreateFingerprint](#)
Creates a fingerprint for the current page, even when there is an existing fingerprint.
- [DetailFix](#)
Calculates the quantity, price, and line total when one of them is blank within a detail field. This calculation is done for all detail lines on the page.
- [FindExportImage](#)
Searches the batch directory for a file that corresponds to the current page that contains the current field. The file extension must match the extension that is specified with the input parameter.
- [FPXMLUsed](#)
Indicates when FPXML format fingerprint files are being used. Call this action when you are using fingerprints from FPXML with Invoice.rrx.

- [GenerateDetails](#)
Sets up a field in APT and puts a detail subfield on every page. This extra detail subfield appears on all pages of a multi-page invoice, allowing every page to be viewed.
- [iloc_SetDetailSimple](#)
Fills the Setup DCO from Runtime Data file.
- [IncrementBatchVar](#)
Increments a batch level variable by 1. When the variable does not exist, it is created. When the variable exists but the value is not a number, the variable is set to 1.
- [IsFingerPrintClass](#)
Connects to the fingerprint database and verifies that the specified fingerprint class contains the fingerprint ID of the current page.
- [IsInINI](#)
Reads and returns to the action the value of the specified key in the INI file.
- [IsInList](#)
Validates that the value of a field is contained within a string.
- [IsMultipageDocument](#)
Determines whether the current object is a document with multiple pages attached.
- [IsSinglePageDocument](#)
Determines whether the current object is a document with only 1 page attached.
- [IsStationIDSuffix](#)
Tests the current station ID by checking that the specified parameter matches the rightmost portion of the station ID.
- [Is_JobNamePrefix](#)
Tests the leftmost portion of the job name to determine if it matches the provided prefix.
- [LoadCCOFromField](#)
Loads the DCO from a field object. The verify panels do not load the CCO into the scripting engine so this action accomplishes that task.
- [PopulateZNLineItemFieldDynamic](#)
This action is like the [PopulateZNLineItemField](#) except that it uses the CCO that was loaded into memory by [LoadCCOFromField](#), instead of the global CCO.
- [ReadFPXMLZones](#)
Reads the zones from the FPXML into the objects for the page and stores the specified fields. When it creates detail lines, it knows the positions of the fields of these detail lines.
- [ScanLineItemDynamic](#)
Scans the line items from the CCO that was loaded into the field. This action is like [ScanLineItem](#) except that it uses the CCO loaded for the field and reads position variables from the line item level.
- [SendOutlookNotification](#)
Uses Outlook to send a notification to specified email addresses. The message within the email is determined by previous calls to actions with a set notification, such as [CheckFreeDiskSpace](#).
- [SetDynamicDetailZones](#)
Takes the position of the line items and builds the line coordinates. It sets the details zones from the first line to the end of the CCO.
- [SetStickyNo](#)
Sets the Sticky indicator to No to indicate that there are no sticky fingerprints. Sticky fingerprints identify a page within a single verification session when another form of the same type appears after a previous form was zoned.
- [SwapImages](#)
Interchanges the TIF for the current page with another TIF that has the same file name but a different extension.
- [SwitchMMDD](#)
Switches US Month and Day date values. It swaps the first 2 characters of the field value with the 2 characters that follow the separator.

- [UpdateFPStats](#)
Updates the fingerprint statistics in the fingerprint database. This action keeps track of the last accessed fingerprint and the number of times a fingerprint is accessed.
- [ValidateVendor](#)
Checks whether the current vendor, vendor number, and postal code exist on the same record in the lookup database.
- [WriteErrorMessage](#)
Writes the message to field level variable message that appears in the status bar. The message is stored in the *MESSAGE* variable.

Parent topic: [Global actions](#)

AddToDetailErrorMsg

Appends extra text to the existing error message.

Syntax

```
bool AddToDetailErrorMsg (StrParam)
```

Parameters

The error message text.

Returns

Always False.

Level

Field level.

Details

Appends additional text to the existing error message.

Example

```
AddToDetailErrorMsg("Description cannot be blank.")
```

Parent topic: [Invoice actions](#)

AddToErrorMsg

Appends the supplied text to any existing error message string.

Syntax

```
bool AddToErrorMsg (StrParam)
```

Parameters

The error message text.

Returns

Always False.

Level

Field level.

Details

Appends the supplied text to any existing error message string.

Example

```
AddToErrorMsg("Vendor Number cannot be blank.")
```

Parent topic: [Invoice actions](#)

AllMixedCase

Changes the value of the current field to be Title case. The first letter of each word is capitalized.

Syntax

```
bool AllMixedCase ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

This action will change the value of the current field to be Title case. The first letter of each word will be capitalized.

Example

```
AllMixedCase ()
```

If the field in this example is "hello, i MUST be going.", the text will be changed to "Hello, I Must Be Going."

Parent topic: [Invoice actions](#)

AlterDatebyDay

Adds the specified number of days to the date contained in the current field. The original character confidence is not changed.

Syntax

```
bool AlterDatebyDay (StrParam)
```

Parameters

The number of days to add.

Returns

False if called at the wrong level or if the input is not numeric. Otherwise, True.

Level

Field level.

Details

This action will add the specified number of days to the date contained in the current field. The original character confidence is not changed.

Example

```
AlterDatebyDay("7")
```

This example adds one week to the date contained in the field. If the date crosses over a month or year boundary, they will be adjusted appropriately.

Parent topic: [Invoice actions](#)

CalculateNotesZone

Creates a zone between detail lines to recognize the text between lines.

Syntax

```
bool CalculateNotesZone ()
```

Parameters

None.

Returns

Always True.

Level

Field level, on the detail field.

Details

This action creates a zone between detail lines, so the text between lines can be recognized.

Example

```
CalculateNotesZone()
```

Parent topic: [Invoice actions](#)

CaptureOpInfo

Captures the Operator, Station ID, and current time and place them into variables. The variables are named based on the provided variable prefix, such as, *prefix Operator*, *prefix Station*, and *prefix Time*. When, a prefix is not provided the task name is used as the prefix by default.

Syntax

```
bool CaptureOpInfo (StrParam)
```

Parameters

An optional prefix to the variable name.

Returns

Always True.

Level

Any level.

Details

This action will capture the Operator, Station ID and current time and place them into variables. The variables are named based on the provided variable prefix, like this: *prefix Operator*, *prefix Station*, and *prefix Time*. If a prefix is not provided, the task name is used as the prefix by default.

Example

```
Status_Preserve_OFF()  
ClearErrorMsg()  
CaptureOpInfo("Production")  
rrCompare("@P\Routing_Instructions", "None")
```

In this example, the variables created will be *Production Operator*, *Production Station*, and *Production Time*.

Parent topic: [Invoice actions](#)

CheckAndFixDecimal

Replaces the space or comma with a decimal point. This action is used to fix errors where the decimal is not recognized and leaves a blank in that area.

Syntax

```
bool CheckAndFixDecimal ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Replaces the space or , with . to fix errors where the decimal is not recognized and leaves a blank in that area. This action can also be used for conversion for European numbers that use a comma to separate dollars from cents so \$100,00 becomes \$100.00 or \$100 00 becomes \$100.00.

Example

```
CheckAndFixDecimal()  
Is_InCharSet("0123456789.,-$")  
AllowOnlyChars("0123456789.-")  
IsFieldCurrency()
```

Parent topic: [Invoice actions](#)

CheckForSticky

For new fingerprints, checks whether there is another matching fingerprint within the same batch that was already verified. It can be used to obtain zone information.

Syntax

```
bool CheckForSticky ()
```

Parameters

None.

Returns

Always True.

Level

Page level.

Details

An example of this action is when a batch contains two similar invoices, which have never been processed by the system before. Once the first invoice is zoned during verify, the same new fingerprint can be used on the second invoice. This is only needed on new matching fingerprints within the same batch because a fingerprints zones are saved at export time and are available for subsequent batches.

Example

```
CheckForSticky()
```

Parent topic: [Invoice actions](#)

CheckFreeDiskSpace

Reads the LowDiskSpaceThreshold setting from the [Notifications] section of the INI file that is specified in the first parameter.

Syntax

```
bool CheckFreeDiskSpace (StrParam)
```

Parameters

Two comma separated parameters:

1. The path to .ini file that contains the LowDiskSpaceThreshold setting.
2. The letter drive of the drive to check. The letter drive must be accompanied by a colon, for example, C:.

Returns

False if called at the wrong level, if parameters are missing or if the settings.ini file cannot be found. Otherwise, True.

Level

Batch level.

Details

If the available disk space is less than the value specified in the .ini file, then a notification will be created in the .ini file. The notification is created under the [Notifications] section of the .ini file, under the LowDiskNotification setting. This setting will be set to Yes.

If the LowDiskSpaceThreshold is missing from the .ini file, the default value of 3000 bytes will be used.

If a notification has been previously created, and the disk space has been increased since the last time the action ran, then the notification will be removed from the .ini file and LowDiskNotification is set to No.

Example

```
CheckFreeDiskSpace ("C:\Datacap\APT\dco_APT\settings.ini,C:")  
Scan()
```

Parent topic: [Invoice actions](#)

ClearErrorMsg

Use this action to clear the current error message.

Syntax

```
bool ClearErrorMsg ()
```

Parameters

None.

Returns

Always True.

Level

Any level.

Details

This action clears the current error message.

Example

```
Status_Preserve_OFF()  
ClearErrorMsg()  
CaptureOpInfo()  
rrCompare("@P\Routing_Instructions", "None")
```

Parent topic: [Invoice actions](#)

CreateFingerprint

Creates a fingerprint for the current page, even when there is an existing fingerprint.

Syntax

```
bool CreateFingerprint ()
```

Parameters

None.

Returns

True if a new fingerprint has been created. Otherwise, False.

Level

Page level.

Details

This action will create a fingerprint for the current page. Forces the creation of a new fingerprint using the current image, even if there is an existing fingerprint. A typical scenario is when there is a very similar fingerprint that incorrectly matches the current image, such as two invoices that are very similar. When the new invoice is received in the future, it should now match on this new fingerprint.

Example

```
IsChildFieldValue("Add_New_Fingerprint,YES")
SetFingerprintDir("@APPPATH(fingerprint)")
SetFingerprintRecogPriority("True")
RecognizePageOCR_S()
CreateFingerprint()
SetFingerprint("@P\Vendor")
iloc_SetDetailSimple("Details")
IncrementBatchVar("Verify requested Add Fingerprint")
```

Parent topic: [Invoice actions](#)

DetailFix

Calculates the quantity, price, and line total when one of them is blank within a detail field. This calculation is done for all detail lines on the page.

Syntax

```
bool DetailFix ()
```

Parameters

None.

Returns

False if called at the wrong level.

False if any of following fields do not exist or are not numeric: Qty, Price, and LineTotal.

Otherwise, True.

Level

Field level.

Details

A typical scenario is a similar fingerprint incorrectly matches the current image, such as two similar invoices. When a new invoice is received in the future, it now matches this new fingerprint.

Example

```
DetailFix()
```

Parent topic: [Invoice actions](#)

FindExportImage

Searches the batch directory for a file that corresponds to the current page that contains the current field. The file extension must match the extension that is specified with the input parameter.

Syntax

```
bool FindExportImage (StrParam)
```

Parameters

One of the following values:

1. Document Level TIFF
2. Document Level PDF
3. Page Level TIO
4. Page Level TIF

Returns

False if called at the wrong level or if the input parameter is invalid. True if the correctly named file with the specified extension exists in the batch directory.

Level

Field level.

Details

Example

```
FindExportImage ("1")
```

Parent topic: [Invoice actions](#)

FPXMLUsed

Indicates when FPXML format fingerprint files are being used. Call this action when you are using fingerprints from FPXML with Invoice.rrx.

Syntax

```
bool FPXMLUsed ()
```

Parameters

None.

Returns

True if a FPXML fingerprint file exist for the current page object. Otherwise, False.

Level

Page level.

Details

Call this action if you are using fingerprints from FPXML with Invoice.rrx.

Example

```
FPXMLUsed()  
ZoneBOTTOM_ImageBottom()  
ScanDetails()
```

Parent topic: [Invoice actions](#)

GenerateDetails

Sets up a field in APT and puts a detail subfield on every page. This extra detail subfield appears on all pages of a multi-page invoice, allowing every page to be viewed.

Syntax

```
bool GenerateDetails ()
```

Parameters

None.

Returns

False if called on the wrong level. Otherwise, True.

Level

Field level.

Details

This action sets up a field in APT and puts a detail subfield on every page. This extra detail subfield appears on all pages of a multi-page invoice, allowing every page to be viewed.

Example

```
GenerateDetails()
```

Parent topic: [Invoice actions](#)

iloc_SetDetailSimple

Fills the Setup DCO from Runtime Data file.

Syntax

```
bool iloc_SetDetailSimple (sDetailName)
```

Parameters

The detail Datacap Object (DCO) Type.

Returns

False if called at the wrong level, if the DCO node cannot be found, if the fingerprint ID is not found, or if the setup DCO cannot be saved. Otherwise, True.

Level

Page level.

Details

This action fills the Setup DCO from Runtime Data file.

Example

```
SetFingerprint("@P\Vendor")
iloc_SetZones()
iloc_SetDetailSimple("Details")
IncrementBatchVar("Intellocate Fingerprint")
```

Parent topic: [Invoice actions](#)

IncrementBatchVar

Increments a batch level variable by 1. When the variable does not exist, it is created. When the variable exists but the value is not a number, the variable is set to 1.

Syntax

```
bool IncrementBatchVar (StrParam)
```

Parameters

The name of the variable to increment.

Returns

Always True.

Level

Any level.

Details

This action increments a batch level variable by 1. If the variable does not already exist, it will be created. If the variable exists but the value is not numeric, the variable is set to 1.

Example

```
rrCompareNot("@P.RecogStatus","1")
SetDocStatus("128")
IncrementBatchVar("Recog - Deleted Document")
```

Parent topic: [Invoice actions](#)

IsFingerPrintClass

Connects to the fingerprint database and verifies that the specified fingerprint class contains the fingerprint ID of the current page.

Syntax

```
bool IsFingerPrintClass (StrParam)
```

Parameters

Two comma separated parameters:

1. The database connection string. Smart parameters are supported.
2. Fingerprint class.

Returns

False if called with the wrong number of parameters, on the wrong level, if the fingerprint ID does not exist in the specified class.

Level

Page Level.

Details

Example

```
ChkDCOStatus("75")
IsFingerPrintClass("@APPVAR(* /fingerprintconn:cs)+, [New]")
DeleteFingerprint()
IncrementBatchVar("Deleted New Fingerprint")
```

A parameter of [New] checks to see if the fingerprint exists in the new class, and if it does, follow on actions can perform further processing.

Parent topic: [Invoice actions](#)

IsInINI

Reads and returns to the action the value of the specified key in the INI file.

Syntax

```
bool IsInINI (StrParam)
```

Parameters

A comma separated string of:

1. The INI Filename.
2. The section within the INI file.
3. The keyword to find in the section.

Returns

True if the value of the current field is a substring within the string specified in the INI file. Otherwise, False.

Level

Field Level.

Details

When possible it is recommended to use the generic actions that read and write INI files in the FileIO action library. Reads and returns to the action the value of the specified key in the INI file. It compares the value of the current field with the with the string in the INI file. The field must be a substring of the string in the INI file.

Example

```
IsInINI("C:\MyDir\settings.ini", "mysection", "mykey")
```

Parent topic: [Invoice actions](#)

IsInList

Validates that the value of a field is contained within a string.

Syntax

```
bool IsInList (StrParam)
```

Parameters

A string that is a substring of the expected Text value.

Returns

True if the value of the *Text* variable of the current field is contained within the specified list.

Level

Field Level.

Details

Example

```
IsInList("Hello Larry")
```

In this example, if the field Text value is *Larry*, the action returns True.

Parent topic: [Invoice actions](#)

IsMultipageDocument

Determines whether the current object is a document with multiple pages attached.

Syntax

```
bool IsMultipageDocument ()
```

Parameters

None.

Returns

False if called at the Batch level, the document object cannot be found, or if the document does not have more than 1 child. Otherwise, True.

Level

Document level.

Details

This action is used to determine if the current object is a document with multiple pages attached.

Example

```
IsMultipageDocument ()
```

Parent topic: [Invoice actions](#)

IsSinglePageDocument

Determines whether the current object is a document with only 1 page attached.

Syntax

```
bool IsSinglePageDocument ()
```

Parameters

None.

Returns

False if not called on a document object or if the document does not contain exactly one page. True if the document contains one page.

Level

Document level.

Details

This action is used to determine if the current object is a document with only 1 page attached.

Example

```
IsSinglePageDocument()  
PopulateZNField()
```

Parent topic: [Invoice actions](#)

IsStationIDSuffix

Tests the current station ID by checking that the specified parameter matches the rightmost portion of the station ID.

Syntax

```
bool IsStationIDSuffix (StrParam)
```

Parameters

The expected Station ID suffix.

Returns

False if the suffix is longer than the current Station ID or if the specified suffix does not match the Station ID. Otherwise, True.

Level

Any level.

Details

This action checks that the specified parameter matches the right most portion of the station ID. This action is useful if you have stations with different suffixes and you want to control actions based on the station name.

Example

```
IsStationIDSuffix("-Test")  
CloseConnection()  
OpenConnection("@APPVAR(*/*lookupdb:cs) ")
```

In this example, If the station name is Validate-Test, the action returns True and continues executing the following actions.

Parent topic: [Invoice actions](#)

Is_JobNamePrefix

Tests the leftmost portion of the job name to determine if it matches the provided prefix.

Syntax

```
bool Is_JobNamePrefix (StrParam)
```

Parameters

The prefix you are expecting for the current job.

Returns

True if the left most portion matches. Otherwise, False.

Level

Any level.

Details

Example

```
Is_JobNamePrefix("Test")
```

If the current job name is TestRecognition, the action returns True.

Parent topic: [Invoice actions](#)

LoadCCOFromField

Loads the DCO from a field object. The verify panels do not load the CCO into the scripting engine so this action accomplishes that task.

Syntax

```
bool LoadCCOFromField ()
```

Parameters

None.

Returns

False if this action is not called on a field or if the CCO file does not exist. Otherwise, True.

Level

Field Level.

Details

This action is required for any invoice action that uses the CCO.

Example

```
LoadCCOFromField()  
SetDynamicDetailZones("Zone Bottom,Notes")  
ZoneBOTTOM_ImageBottom()  
ScanDetails()
```

Parent topic: [Invoice actions](#)

PopulateZNLineItemFieldDynamic

This action is like the `PopulateZNLineItemField` except that it uses the CCO that was loaded into memory by `LoadCCOFromField`, instead of the global CCO.

Syntax

```
bool PopulateZNLineItemFieldDynamic ()
```

Parameters

None

Returns

True if there is no field position or if the data was found in a zone. False if there is a zone defined but it is blank.

Level

Page level.

Details

This action populates the data file of the Fingerprint with the recognized value contained inside the zone of a child Field object of a LINEITEM parent field. This action should only be used with sub-fields of the LINEITEM field - ItemID, ItemDesc, Quantity, Price in the Invoices application.

Example

```
FPXMLUsed()  
PopulateZNLineItemFieldDynamic()
```

Parent topic: [Invoice actions](#)

ReadFPXMLZones

Reads the zones from the FPXML into the objects for the page and stores the specified fields. When it creates detail lines, it knows the positions of the fields of these detail lines.

Syntax

```
bool ReadFPXMLZones (StrParam)
```

Parameters

Comma-separated parameters:

1. The fingerprint directory. Smart parameters are supported.
2. A comma-separated list of the detail lines.

Returns

False if called at the wrong level or if the FPXML does not exist. Otherwise, True.

Level

Page level.

Details

Example

```
ReadFPXMLZones("@APPPATH(fingerprint), details, lineid, itemdesc, qty, price, linetotal")
rrSet("FPXML", "@P.ZoneRead")
IsMultipageDocument()
SetEOL("|")
```

Parent topic: [Invoice actions](#)

ScanLineItemDynamic

Scans the line items from the CCO that was loaded into the field. This action is like ScanLineItem except that it uses the CCO loaded for the field and reads position variables from the line item level.

Syntax

```
bool ScanLineItemDynamic (StrParam)
```

Parameters

A comma separated list of any fields that should be ignored.

Returns

Always True.

Level

Field Level.

Details

This action is required for Find Details functionality. It saves the positions of the line and of the fields at the detail level so all of the line items can be erased and recreated.

Example

```
FPXMLUsed()  
ScanLineItemDynamic("ZoneBottom,Notes")
```

Parent topic: [Invoice actions](#)

SendOutlookNotification

Uses Outlook to send a notification to specified email addresses. The message within the email is determined by previous calls to actions with a set notification, such as [CheckFreeDiskSpace](#).

Syntax

```
bool SendOutlookNotification (StrParam)
```

Parameters

The extension of the attachment. The attachment is expected to be named the same as the current document ID.

Returns

False if called at the wrong level, if the connection to Outlook could not be established, or if email addresses could not be found in [Settings.ini](#).

Level

Document level.

Details

This action requires Outlook to be installed on the computer and logged in with an ID that has the appropriate permissions to send emails.

Example

```
SendOutlookNotification(".pdf")
```

Parent topic: [Invoice actions](#)

Related reference:
[CheckFreeDiskSpace](#)

SetDynamicDetailZones

Takes the position of the line items and builds the line coordinates. It sets the details zones from the first line to the end of the CCO.

Syntax

```
bool SetDynamicDetailZones (StrParam)
```

Parameters

The parameter is the Zone Bottom field. If more than one field is listed, Zone Bottom must be the first field and other fields are ignored.

Returns

False, if no children exist. Otherwise, True.

Level

Page level.

Details

Example

```
LoadCCOFromField()  
SetDynamicDetailZones("Zone Bottom,Notes")  
ZoneBOTTOM_ImageBottom()  
ScanDetails()
```

Parent topic: [Invoice actions](#)

SetStickyNo

Sets the Sticky indicator to No to indicate that there are no sticky fingerprints. Sticky fingerprints identify a page within a single verification session when another form of the same type appears after a previous form was zoned.

Syntax

```
bool SetStickyNo ()
```

Parameters

None.

Returns

Always True.

Level

Any level.

Details

Example

```
IsCurrentObjVariable("Sticky,Yes")  
SetStickyNo()
```

Parent topic: [Invoice actions](#)

SwapImages

Interchanges the TIF for the current page with another TIF that has the same file name but a different extension.

Syntax

```
bool SwapImages (StrParam)
```

Parameters

Two extensions:

1. The extension to be saved as TIF.
2. The current tif file's extension.

Returns

False if called at the wrong level, if the wrong number of parameters are specified, or if the original file cannot be found. True if the file extensions are interchanged.

Level

Page level.

Details

One possible use for this action is if you want to export the original version of the TIF file instead of the converted version of the image used to process the document. The original version of the TIF file had previously been renamed using the extension .TIO (where the O stands for *Original*). So in the example, the SwapImages action will be used to change the current .TIF file to instead have an extension of .TIB (where the B can stand for *Backup*), and then take the original file (.TIO) and give it the .TIF extension.

Example

```
SwapImages ("TIB,TIO")
```

The action renames the current image TM00000n.TIF to TM00000n.TIB. It then copies TM00000n.TIO to TM00000n.TIF and that becomes the new current image.

Parent topic: [Invoice actions](#)

SwitchMMDD

Switches US Month and Day date values. It swaps the first 2 characters of the field value with the 2 characters that follow the separator.

Syntax

```
bool SwitchMMDD (StrParam)
```

Parameters

A list of separators.

Returns

True if two separators are found and the separators are swapped. Otherwise, False.

Level

Field level.

Details

Example

```
SwitchMMDD("/")
```

In this example, the value "03/09/10" becomes "09/03/10".

Parent topic: [Invoice actions](#)

UpdateFPStats

Updates the fingerprint statistics in the fingerprint database. This action keeps track of the last accessed fingerprint and the number of times a fingerprint is accessed.

Syntax

```
bool UpdateFPStats ()
```

Parameters

None.

Returns

False if called at the wrong level or if the fingerprint does not exist. Otherwise, True.

Level

Page level.

Details

Example

```
UpdateFPStats ()
```

Parent topic: [Invoice actions](#)

ValidateVendor

Checks whether the current vendor, vendor number, and postal code exist on the same record in the lookup database.

Syntax

```
bool ValidateVendor (StrParam)
```

Parameters

The connection string to the lookup database. Smart Parameters are supported.

Returns

False if the connection cannot be opened or if the settings file cannot be found.

False if these fields cannot be found: Vendor, Vendor_Number, or Remittance_Zip.

False if the vendor cannot be found in the database.

Otherwise, True.

Level

Page level.

Details

The fields that are validated are in the validate vendor check are Vendor, Vendor_Number, and Remittance_Zip.

Example

```
SetIsOverrideable("False")
ValidateVendor("@APPVAR(*/*lookupdb:cs) ")
```

Parent topic: [Invoice actions](#)

WriteErrorMessage

Writes the message to field level variable message that appears in the status bar. The message is stored in the *MESSAGE* variable.

Syntax

```
bool WriteErrorMessage (StrParam)
```

Parameters

The error message.

Returns

Always False.

Level

All levels.

Details

Example

```
GetDCOStatus("75")
WriteErrorMessage("The page is set to a deleted status.")
```

Parent topic: [Invoice actions](#)

IOverlay actions

Use the IOverlay actions to combine the current page image with a background image. You can use this action to reapply a form background that dropped out during scanning.

The IOverlay actions enable dithering of a background image and haloing with white pixels around the black pixels on an image. These actions make the image easier to read.

- [Overlay](#)
Combines the current image with the image file specified by the SetBackgroundImage action into a new image replacing the current image. This action is used to reinstate a form background that was dropped out during scanning.
- [SetBackgroundImage](#)
Designates the Image file that will overlay the image of the current page.
- [SetDitheringBackground](#)
Enables or disables dithering of the background image. Dithering makes the background image appear lighter than the information of the current image so that visually it appears less prominent.
- [SetHaloBackground](#)
Enables or disables a halo of white pixels around any black pixels from the current image where they would otherwise touch pixels from the background. This makes the foreground information easier to read.

Parent topic: [Global actions](#)

Overlay

Combines the current image with the image file specified by the SetBackgroundImage action into a new image replacing the current image. This action is used to reinstate a form background that was dropped out during scanning.

Syntax

```
bool Overlay ()
```

Parameters

None.

1. The parameter of the preceding SetBackgroundImage action can be a smart parameter that locates a file.
2. Looks for a field type called Image_Offset or IMAGEOVERLAYOFFSET containing a comma separated value containing the amount of X and Y pixel offset for the overlay image.

These fields are automatically generated by running FindFingerprint, CalculateOffset, WordFind_Offset, CalculateLocalOffset, Autofield(), and most of the PatternMatch.rxx actions.

Returns

False if the action is not applied to a Page object; if it cannot locate the Background Image file; or if the action encounters an error of a different kind. Otherwise, True.

Level

Page level only.

Details

Example:

```
SetBackgroundImage (c:\ParentDir\mclaims\process\hcfa\hcfa.tif)
Overlay()
```

In this example, the Overlay action uses the file name and path in the parameter of the SetBackgroundImage action to locate the Background Image file.

```
SetBackgroundImage (@APPPATH(formdir)+\+ub04.tif)
Overlay()
```

Here, the SetBackgroundImage action uses a smart parameter to load the file name with the directory path coming from the application service.

Note: SetBackgroundImage must be called before Overlay. Other actions such as SetHaloBackground and SetDithering are optional and, if used, should precede the call to the Overlay action.

Parent topic: [IOverlay actions](#)

SetBackgroundImage

Designates the Image file that will overlay the image of the current page.

Syntax

```
bool SetBackgroundImage (StrParam)
```

Parameters

Full path to the overlay Image file.

Returns

False if image does not exist, otherwise True.

Level

All.

Details

Designates the Image file that will overlay the image of the current page. Smart Parameters supported.

Example:

```
SetBackgroundImage(c:\ParentDir\mclaims\process\hcfa\hcfat.tif)
SetDitheringBackground(True)
SetHaloBackground(True)
Overlay()
```

Parent topic: [IOverlay actions](#)

SetDitheringBackground

Enables or disables dithering of the background image. Dithering makes the background image appear lighter than the information of the current image so that visually it appears less prominent.

Syntax

```
bool SetDitheringBackground (StrParam)
```

Parameters

String value to enable dithering (True) or disable dithering (False).

Returns

Always True.

Level

All.

Details

Example:

```
SetBackgroundImage(c:\ParentDir\mclaims\process\hcfa\hcfat.tif)
SetDitheringBackground(True) SetHaloBackground(True)
Overlay()
```

Parent topic: [IOverlay actions](#)

SetHaloBackground

Enables or disables a halo of white pixels around any black pixels from the current image where they would otherwise touch pixels from the background. This makes the foreground information easier to read.

Syntax

```
bool SetHaloBackground (StrParam)
```

Parameters

String value to enable a halo (True) or prevent a halo (False).

Returns

Always True.

Level

All.

Details

Example:

```
SetBackgroundImage(c:\ParentDir\mclaims\process\hcfa\hcfat.tif)
SetDitheringBackground(True)
SetHaloBackground(True)
Overlay()
```

Parent topic: [IOverlay actions](#)

Locate actions

Use the Locate actions in combination with full text recognition to locate words or regular expressions on the page. And to move around the page by line or word.

The Locate library also includes a few format validation actions, such as `IsCurrency` and `IsDateValue`. Most of the validation actions are located in the `Validate` library.

- [AddKeyList](#)
Adds a list of keywords or phrases that can be used for matching.
- [AggregateKeyList](#)
Toggles the Key list searches to merge all words or phrases in the list into a single query.
- [CreateVirtualPage](#)
This action creates a new page that is a subset of an existing page or a subset of merged pages.
- [CreateVirtualZone](#)
Creates a field zone at runtime based on the identified text locations.
- [DefaultValue](#)
Sets the text value of the current field in the page data file to the value that is specified.
- [FilterIt](#)
Removes all instances of the specified characters from the word that is located.
- [FindDBList](#)
Locates a word that matches one of a list of words that are obtained from a SQL query.
- [FindDBList_InZone](#)
Locates a word that matches a word from the current field.
- [FindKeyList](#)
Locates the first or next occurrence of a word or phrase that matches one of the entries in a keyword file.
- [FindKeyList_InZone](#)
Locates the first (or next) occurrence of a word or phrase that matches one of the entries in the current field.
- [FindLastKeyList](#)
Locates the last occurrence of a word or phrase that matches one of the entries in a keyword file.
- [FindLastKeyList_InZone](#)
Locates the last occurrence of a word or phrase that matches one of the current field.
- [FindLastRegEx](#)
Same as the `FindLastWord` action, except that it supports regular expressions.

- [FindLastRegEx_InZone](#)
Same as the FindLastWord_InZone action, except that it supports regular expressions.
- [FindLastRegExList](#)
Same as the FindLastKeyList action, except that it supports regular expressions.
- [FindLastRegExList_InZone](#)
Same as the FindLastRegExList action, except that it searches the current field only.
- [FindLastWord](#)
Locates the last occurrence of the specified word or phrase on the current page.
- [FindLastWord_InZone](#)
Locates the last occurrence of the specified word or phrase on the current field.
- [FindNextDBList](#)
Same as the FindDBList action, except that it locates the next instance.
- [FindNextDBList_InZone](#)
Same as the FindNextDBList action, except that it searches the current field only.
- [FindNextKeyList](#)
Same as the FindKeyList action, except that it locates the next instance.
- [FindNextKeyList_InZone](#)
Same as the FindNextKeyList action, except that it searches the current field only.
- [FindNextRegExList](#)
Same as the FindRegExList action, except that it locates the next instance.
- [FindNextRegExList_InZone](#)
Same as the FindNextRegExList action, except that it searches the current field only.
- [FindRegExList](#)
Same as the FindKeyList action, except that it supports regular expressions.
- [FindRegExList_InZone](#)
Same as the FindRegExList action, except that it searches the current field only.
- [GetSelectedBlockType](#)
Obtains the type of the currently selected block and stores it in the DCO location indicated by the target parameter. This action uses the document block layout that is created by the DocumentAnalytics actions.
- [GoAboveWord](#)
Moves up the specified number of lines from the previously found word or phrase.
- [GoBelowWord](#)
Moves down the specified number of lines from the previously found word or phrase.
- [GoDownLine](#)
Moves down the specified number of lines from the previously found word or phrase and selects the first word.
- [GoFirstLine](#)
Moves to the first line of the page when you are running full page recognition.
- [GoFirstWord](#)
Moves to the first word on the current line.
- [GoLastLine](#)
Moves to the last line of the page when you are running full page recognition.
- [GoLastWord](#)
Moves to the last word on the current line.
- [GoLeftWord](#)
Moves the specified number of words to the left of the previously found word or phrase.
- [GoRightWord](#)
Moves the specified number of words to the right of the previously found word or phrase.
- [GoSiblingBlockNext](#)
The currently selected block in the document layout is changed to the next sibling block at the same level in the structure. The number of siblings that are moved is determined by the input parameter. This action uses the document block layout that is created by the DocumentAnalytics actions.

- [GoSiblingBlockPrevious](#)
The currently selected block in the document layout is changed to the previous sibling block at the same level in the structure. The number of siblings moved is determined by the input parameter. This action uses the document block layout that is created by the DocumentAnalytics actions.
- [GoUpLine](#)
Moves up the specified number of lines from the previously found word or phrase and selects the first word.
- [GroupWords](#)
Groups words to the left and right of the previously found word if they are no more than the specified number of character widths apart.
- [GroupWordsLEFT](#)
Groups words to the left of the previously found word if they are no more than the specified number of character widths apart.
- [GroupWordsRIGHT](#)
Groups words to the right of the previously found word if they are no more than the specified number of character widths apart.
- [IsAlpha](#)
Determines whether the specified percentage of characters in a located word is letters (defaults to 100%)
- [IsCurrency](#)
Determines whether the value of the located word is a currency value. The value contains numbers and includes a two-digit decimal amount.
- [IsDateValue](#)
Determines whether the value of the located word is in one of the supported date formats.
- [IsNumber](#)
Determines whether the specified percentage of characters in a located word is numbers (defaults to 100%)
- [IsSelectedBlockType](#)
Compares the provided block type to the type of the currently selected block. This action uses the document block layout that is created by the DocumentAnalytics actions.
- [IsValue](#)
Determines whether the value of the located word matches the value that is specified.
- [IsValue_RegEx](#)
Determines whether the value of the located word matches the regular expression specified.
- [LocatePositionRestore](#)
LocatePositionRestore returns the internal locate word pointer to the word position it was on when LocatePositionSave was called.
- [LocatePositionSave](#)
LocatePositionSave creates a restore point for the current word position of the locate pointer.
- [MaxLength](#)
Determines whether the number of characters in the located word is greater than or equal to the number specified.
- [MergeWordLF](#)
Merges the located word with one or more words to the left, on the same line.
- [MergeWordRT](#)
Merges the located word with one or more words to the right, on the same line.
- [MinLength](#)
Determines whether the number of characters in the located word is less than or equal to the number specified.
- [RegExFind](#)
Same as the WordFind action, except that it supports regular expressions.
- [RegExFind_InBlock](#)
RegExFind_InBlock will look for the first occurrence of a word or phrase that matches the regular

expression. This action uses the document block layout that is created by the DocumentAnalytics actions.

- [RegExFind_InZone](#)
Same as the RegExFind action, except that it searches the current field only.
- [RegExFindNext](#)
Same as the WordFindNext action, except that it supports regular expressions.
- [RegExFindNext_InBlock](#)
Starting from the location of a previously found word or phrase, this action locates the first occurrence of a word or phrase in the currently selected block where the input search term is specified as a regular expression. Once the regular expression has been matched, the location of the found word or phrase will be remembered so the result can be used by subsequent actions. This action uses the document block layout that is created by the DocumentAnalytics actions.
- [RegExFindNext_InZone](#)
Same as the RegExFindNext action, except that it searches the current field only.
- [ScanRT](#)
Moves the specified number of words to the right of the current word. Expands the search area up and down slightly in case the word is a little above or below the current word.
- [SelectParentBlock](#)
Looks for the first occurrence of a word or phrase that matches the regular expression. If a match is found, the next action will change the currently selected block from the found word to the entire line that contains the word. SelectParentBlock is then called again to change the selected block to the parent of the current line, then the current field zone is updated to contain the entire set of lines contained in the current block and the field value is set to all of the text within the current block. This action utilizes the document block layout created by the DocumentAnalytics actions.
- [SelectParentBlockOuterType](#)
Selects the outermost parent of the current block of the type that is specified. If the currently selected block has multiple parent blocks of the same type, then the parent farthest from the current block is selected. This action uses the document block layout that is created by the DocumentAnalytics actions.
- [SelectParentBlockType](#)
Selects the parent of the current block of the type that is specified. If the currently selected block has multiple parent blocks of the same type, then the parent nearest to the current block would be selected. Uses the document block layout created by the DocumentAnalytics actions.
- [SelectSnippet](#)
Populates a snippet field with the recognized value of the located word. Used with directional actions.
- [SetKeyFileEncodingAsUnicode](#)
Determines if the Locate key files are opened as ASCII or UNICODE.
- [SetRect](#)
Sets the position and size of the current field in the page data file to the values specified.
- [SetVirtualPageEndPosition](#)
Use this action after you use the Locate commands that identify the last word of a new virtual page.
- [SetVirtualPageStartPosition](#)
Use this action after you use the Locate commands that identify the first word of a new virtual page.
- [UpdateDCOField](#)
Updates the position coordinates for the specified field in the page data file with the position of the located word.
- [UpdateField](#)
Updates the current field in the page data file with the value and position of the located word.
- [UpdateFieldWithBlock](#)
Updates the current DCO object's field zone coordinates with the coordinates of the currently selected block. The text of the field will be set with the contents of the text bound by the currently selected block in the document layout. Uses the document block layout created by the DocumentAnalytics actions.
- [ValueInField](#)
Determines whether any part of the located word matches the value that is specified.

- [ValueInField_Fuzzy](#)
Uses fuzzy matching to determine whether any part of the located word matches the value that is specified.
- [ValueInField_RegEx](#)
Determines whether any character or series of characters in the located word matches the specified regular expression.
- [WordFind](#)
Locates the first or next occurrence of the specified word or phrase on the current page.
- [WordFind_InZone](#)
Same as the WordFind action, except that it searches the current field only.
- [WordFindNext](#)
Same as the WordFind action, except that it locates the next occurrence.
- [WordFindNext_InZone](#)
Same as the WordFindNext action, except that it searches the current field only.
- [WordFind_Offset](#)
Sets the value of the page's Image_Offset variable. The variable is based on the difference in the position of the specified word on the current page and on the matched fingerprint image.

Parent topic: [Global actions](#)

AddKeyList

Adds a list of keywords or phrases that can be used for matching.

Syntax

```
bool AddKeyList (string KeyName, string KeyWord_01, string KeyWord_02, string
KeyWord_03, string KeyWord_04, string KeyWord_05, string KeyWord_06, string
KeyWord_07, string KeyWord_08, string KeyWord_09, string KeyWord_10, string
KeyWord_11, string KeyWord_12, string KeyWord_13, string KeyWord_14, string
KeyWord_15, string KeyWord_16, string KeyWord_17, string KeyWord_18, string
KeyWord_19, string KeyWord_20, string KeyWord_21, string KeyWord_22, string
KeyWord_23, string KeyWord_24, string KeyWord_25)
```

Parameters

1. String value used as a reference name for other actions to call this list of keywords or phrases.
2. Up to twenty five keywords or phrases to be used for matching.

Returns

False if no keywords or phrases are entered. Otherwise, True.

Level

Any.

Details

Adds a list of keywords or phrases that can be used for matching.

This action complements the List search actions that can load a Keyword text file which contains a list of words or phrases, separated by new lines, that are used for matching.

AddKeyList adds a list of up to 25 words or phrases without having to read or edit a file.
Attention: You can not redefine or overwrite an existing key list

Example

```
AddKeyList("InvNum","Invoice Number","Inv. Num.,""Invoice #:")  
FindKeyList("InvNum")
```

This action searches the current page, from first word to the last word of the current page, for the first occurrence of all keywords in the Invoices Number Keyword list "InvNum".

If successful, the search stops and remembers the location of that word for subsequent actions.

If not successful, the action continues searching for the next keyword in the InvNum list, starting from first word of the current page, repeating this search pattern until a match is found, or until there are no more keywords.

Parent topic: [Locate actions](#)

Related reference:

[FindKeyList](#)

[FindKeyList_InZone](#)

[FindNextKeyList](#)

[FindNextKeyList_InZone](#)

[FindRegExList](#)

[FindRegExList_InZone](#)

[FindNextRegExList](#)

[FindNextRegExList_InZone](#)

[FindLastKeyList](#)

[FindLastKeyList_InZone](#)

[FindLastRegEx](#)

[FindLastRegEx_InZone](#)

[AggregateKeyList](#)

AggregateKeyList

Toggles the Key list searches to merge all words or phrases in the list into a single query.

Syntax

```
bool AggregateKeyList (bool SearchAll)
```

Parameters

Toggles Key list searches to merge all words or phrases in the list into a single query.

Returns

Always True.

Level

Any.

Details

Use of this action changes the search behavior for all 'List' type key word search actions. Default behavior is to search each keyword in the list sequentially. This action with the parameter set to True changes the search behavior to check for all the keywords in a single aggregate keyword search. To change back to the default behavior use this action with the False parameter.

Example

```
AggregateKeyList (True)  
FindKeyList ("InvNum")
```

This action searches the current page, from first word to the last word of the current page, for the first occurrence of all keywords in the Invoices Number Keyword file (InvNum.key). If successful, the search stops and remembers the location of that word for subsequent actions; if not, the action continues searching for the next word in InvNum.key, starting from first word of the current page, repeating this search pattern until a match is found, or until there are no more keywords.

Parent topic: [Locate actions](#)

Related reference:

[FindKeyList](#)

[FindNextKeyList](#)

[FindLastKeyList](#)

[AddKeyList](#)

CreateVirtualPage

This action creates a new page that is a subset of an existing page or a subset of merged pages.

Syntax

```
bool CreateVirtualPage (string sPageType)
```

Parameters

The new page type for the created page can be provided as a parameter to this action. The provided name is assigned as the type for the new DCO page that is created by this action. If no parameter is provided, the new page type is default to "Other". Smart parameters are supported.

Returns

True, if the new page is created. False, if an error occurs or if the start and end points for the block are not set.

Level

Page level.

Details

Using the actions SetVirtualPageStartPosition and SetVirtualPageEndPosition, a segment of text from within the current page can be duplicated into a unique page. This approach can be useful to separate out a required area of text to allow subsequent actions to work solely on the required area.

A layout file is needed to use this action and the CCO must be loaded from the layout file by using the CreateCcoFromLayout action. The recognition of the layout file and creation of the CCO for the page might be performed in the same ruleset or in an earlier ruleset. Multiple pages within the same document can be merged

together by using the action MergeLayoutByType action to create a multi-page segment of text to be extracted into a new page. For example, it is possible to merge together three pages of text. You can create a new page that starts with text from the bottom of page 1, all of the text in page 2, and text from the start of page 3. When you create a new page, all of the text must be continuous.

It is possible to create multiple new page subsets from a single page or single merged page. Use the Start and End actions to define the start and end of a page and create the new virtual page. Then, define new start and end points, create another new page, and so on.

The new pages that are created are no different than any other page. Actions that work on a page object can be used on the new pages, allowing processing on these new pages as needed. For example, these new pages can be combined into their own new document and then processed with extra searches or exported.

When a new page is created, the image for the current page is copied to the name for the new page object. The new page object is then added to the application's runtime DCO. The new page ID and associated image, and layout file use the name of the existing page with an underscore added and a number. This number is incremented if CreateVirtualPage is called multiple times on the same page. For example, "TM000001_0". CreateCcoFromLayout can be called on the new page and have extra rules run on it as needed.

Example

```
Recognize ()
CreateCcoFromLayout ()
WordFind ("Hello")
SetVirtualPageStartPosition ()
WordFindNext ("Good bye")
GoUpLine ("1")
GoLastWord ()
SetVirtualPageEndPosition ()
CreateVirtualPage ("MyNewPageType")
```

This example recognizes a page, creating a layout file. The CCO is created from the layout file and then Locate actions are used to find the first word to include in the new page. The start of the new page's text segment is identified by calling SetVirtualPageStartPosition. The end of the text segment is located. In this example, the word "Good bye" indicates it is the end of the desired text. Because the word "Good bye" should not be included in the new page, the located word position is moved to the last word of the line above the word "Good bye". SetVirtualPageEndPosition then identifies this new position as the end of the new page segment. Finally, the action CreateVirtualPage is called to create the new page. After this action completes, the new page will exist in the document hierarchy and can be acted upon by subsequent rules and actions.

Parent topic: [Locate actions](#)

CreateVirtualZone

Creates a field zone at runtime based on the identified text locations.

Member of namespace

Locate

Syntax

```
bool CreateVirtualZone (string sFieldName)
```

Parameters

sFieldName

Type: String

Name of the field that has its zone assigned based on the previously set virtual start and end positions. If the field does not exist, it is created. Smart parameters are supported.

Returns

True if the zone is set on the field. Otherwise, returns False if there is an error or if the start and end points for the zone are not set.

Level

Page or Field level.

Details

This action assigns zone coordinates to a field based on previously set start and end positions that were identified using locate actions. Using the actions SetVirtualPageStartPosition and SetVirtualPageEndPosition, a rectangular segment of text from within the current page can be identified at run time and set as a zoned area. This approach can be useful to set the zone position of a field that is not a fixed location but can be determined by searching the recognized text to find the top-left corner and bottom-right corner of the zone area.

A layout file is required to use this action and the CCO must be loaded from the layout file by using the CreateCcoFromLayout action. The recognition of the layout file and creation of the CCO for the page can be performed in the same ruleset or in an earlier ruleset.

Once the zone is set, it can be used just like a zone that is created from a fingerprint. Subsequent actions can be used to process the zoned text.

A field with coordinates set based on the contents of the recognized page can have many uses. A less obvious use is for table identification when the table boundary is not determined by grid lines or the engine is not properly determining the table area even if grid lines exist. In this scenario, use the locate actions to determine the top-left and bottom-right coordinates of the table with steps similar to the example below. This will find the table location based on the text that exists on the page. There would need to be expected text or positions to identify the table boundaries. After calling CreateVirtualZone, then the field can be identified with the variable `y_TableZone` as described in the help for the OCR/A Recognize action to provide table boundaries for better table identification by making a second call to the Recognize action.

Example:

```
Recognize ()
CreateCcoFromLayout ()
WordFind ("Hello")
SetVirtualPageStartPosition ()
WordFindNext ("Good bye")
GoUpLine ("1")
GoLastWord ()
SetVirtualPageEndPosition ()
CreateVirtualZone ("MyField")
```

This example recognizes a page, creating a layout file. The CCO is created from the layout file and then Locate actions are used to find the first word to include in the new page. The start of the new page's text segment is identified by calling SetVirtualPageStartPosition. This location is used at the left-top coordinate for the zone. The end of the text segment is located. In this example, the word "Good bye" indicates it is the end of the desired text. Because the word "Good bye" should not be included in the

new page, the located word position is moved to the last word of the line above the word "Good bye". The SetVirtualPageEndPosition then identifies this new position as the bottom-right position for the zone. Finally, the action CreateVirtualZone is called to set the zone for the specified field based on the found text.

Parent topic: [Locate actions](#)

DefaultValue

Sets the text value of the current field in the page data file to the value that is specified.

Syntax

```
bool DefaultValue (strParam)
```

Parameters

The String value assigned to the bound Field object of the Document Hierarchy that represents the current field.

Returns

Always True.

Level

Field level only.

Details

Sets the captured value of the current field to the String value you enter as a parameter.

Example:

```
DefaultValue("Bill Paid")
DefaultValue("Past Due")
```

Parent topic: [Locate actions](#)

FilterIt

Removes all instances of the specified characters from the word that is located.

Syntax

```
bool FilterIt (StrParam)
```

Parameters

A string containing the characters to remove.

Important: This action removes every instance of the character or characters.

Returns

Always True.

Level

Field level only.

Details

Removes all instances of each character you enter as a parameter from the located word or phrase.

Example

```
FilterIt("-")
31-Dec-01 becomes 31Dec01
FilterIt("1")
31-Dec-01 becomes 3-Dec-0
FilterIt("1-")
31-Dec-01 becomes 3Dec0
```

Parent topic: [Locate actions](#)

FindDBList

Locates a word that matches one of a list of words that are obtained from a SQL query.

Syntax

```
bool FindDBList (StrParam)
```

Parameters

An SQL statement whose result returns a word or phrase, or a list of words or phrases.

Returns

True if the word or phrase is located on the page. Otherwise, False.

Level

Page or field level.

Details

This action issues an SQL query against the lookup database, returning a list of words or phrases that will be located on the current page. The search is performed from the first word of the current page. Initially, the first listed word or phrase is searched for on the page. If there is no match, it will search for the next word or phrase returned from the database. This continues until a match is found or none of the returned words are contained on the page. The location of the found word or phrase will be remembered so the result can be used by subsequent actions. The search is case sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes "will" and the recognition read "wi11", a match will still occur.

Common substitutions include characters: B8, Z2, S5, oO0 and iItl1.

Example

```
FindDBList("Select LastName from Providers")
```

This example gets a list of last names from a provider table and find the first matching occurrence on the page.

Parent topic: [Locate actions](#)

Related reference:

[FindDBList_InZone](#)

[FindNextDBList](#)

FindDBList_InZone

Locates a word that matches a word from the current field.

Syntax

```
bool FindDBList_InZone (StrParam)
```

Parameters

An SQL statement whose result returns a word or phrase, or a list of word or phrases.

Returns

True, if the word or phrase is on the page. Otherwise, False.

Level

Page or field level.

Details

This action generates an SQL query against the Lookup database, returning a list of words or phrases that are in the current field. The search is performed from the first word of the current field. Initially, the first listed word or phrase is searched for in the field. If there is no match, it will search for the next word or phrase that is returned from the database. The search continues until a match is found or none of the returned words are contained on the page. The location of the found word or phrase is remembered so the result can be used by subsequent actions. The search is case-sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes `will` and the recognition that is read is `wi11`, a match still occurs.

Common substitutions include characters: B8, Z2, S5, o00, and iTt1.

Example

```
FindDBList("Select LastName from Providers")
```

This example obtains a `LastName` list from a provider table and finds the first matching occurrence on the page.

Parent topic: [Locate actions](#)

Related reference:[FindNextDBList](#)[FindNextDBList_InZone](#)

FindKeyList

Locates the first or next occurrence of a word or phrase that matches one of the entries in a keyword file.

Syntax

```
bool FindKeyList (StrParam)
```

Parameters

The name of the Keyword text file. The file will contain a list of words or phrases, separated by new lines, that will be used for matching.

The file name can be provided in one of two ways:

1. A full path name of the file, including the .key extension.
2. The file name only, with no extension specified. The application will look in the process directory and the file must have a .key extension.

Returns

True if at least one word on the page matches any word or pattern in the Keyword file. Otherwise, False.

Level

Field level only.

Details

Opens the Keyword file you specify as a parameter, then checks the words or phrases on the current page against the keywords in the list to find a match. The location of the found word or phrase that matches an entry in the keyword file will be remembered so the result can be used by subsequent actions. Matching is case sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes "will" and the recognition read "wi11", a match will still occur.

Common substitutions include characters: B8, Z2, S5, o00 and iIt1.

Example

```
FindKeyList ("InvNum")
```

This action searches the current page, from first word to the last word of the current page, for the first keyword in the Invoices Number Keyword file (InvNum.key). If successful, the search stops and remembers the location of that word for subsequent actions; if not, the action continues searching for the next word in InvNum.key, starting from first word of the current page, repeating this search pattern until a match is found, or until there are no more keywords.

Parent topic: [Locate actions](#)

Related reference:

[FindNextKeyList](#)
[FindLastKeyList](#)
[AddKeyList](#)

FindKeyList_InZone

Locates the first (or next) occurrence of a word or phrase that matches one of the entries in the current field.

Syntax

```
bool FindKeyList_InZone (StrParam)
```

Parameters

The name of the Keyword text file. The file will contain a list of words or phrases, separated by new lines, that will be used for matching.

The file name can be provided in one of two ways:

1. A full path name of the file, including the .key extension.
2. The file name only, with no extension specified. The application will look in the process directory and the file must have a .key extension.

Returns

True if at least one word in the current bound zone matches any word or pattern in the Keyword file. Otherwise, False.

Level

Field level.

Details

Opens the Keyword file you specify as a parameter, then checks the words or phrases in the current field of the source page's .cco file against the keywords in the key file list to find a match. The location of the found word or phrase that matches an entry in the keyword file will be remembered so the result can be used by subsequent actions. Matching is case sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes "will" and the recognition read "wi11", a match will still occur.

Common substitutions include characters: B8, Z2, S5, oO0 and iItl1.

Example

```
FindKeyList_InZone("Line_Items")
```

This action searches the current bound zone for the first keyword in the Keyword file (Line_Items.key). If successful, the search stops and remembers the location of that word for subsequent actions; if not, the action continues searching for the next word in Line_Items.key, from the beginning of the current bound zone, repeating this search pattern until a match is found, or until there are no more keywords.

Parent topic: [Locate actions](#)

Related reference:

[FindNextKeyList](#)
[FindLastKeyList](#)
[AddKeyList](#)

FindLastKeyList

Locates the last occurrence of a word or phrase that matches one of the entries in a keyword file.

Syntax

```
bool FindLastKeyList (StrParam)
```

Parameters

The name of the Keyword text file. The file will contain a list of words or phrases, separated by new lines, that will be used for matching.

The file name can be provided in one of two ways:

1. A full path name of the file, including the .key extension.
2. The file name only, with no extension specified. The application will look in the process directory and the file must have a .key extension.

Returns

True if at least one word on the page matches any word or pattern in the Keyword file. Otherwise, False.

Level

Page level.

Details

Opens the Keyword file you specify as a parameter, then checks the words or phrases on the current page against the keywords in the list to find a match. The current page will be searched to find the last occurrence of word or phrase. The location of the last occurrence word or phrase that matches an entry in the keyword file will be remembered so it can be utilized by subsequent actions. Word matching is case sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes "will" and the recognition read "wi11", a match will still occur.

Common substitutions include characters: B8, Z2, S5, o00 and iIt1.

Example

```
FindLastKeyList ("InvNum")
```

This action searches the current page for the last instance of a match with the first keyword in the Keyword file (InvNum.key). If successful, the search stops and remembers the location of that word for subsequent actions; if not, the action continues searching using the next word in InvNum.key, matching the last occurrence of the word, repeating this search pattern until a match is found, or until there are no more keywords.

Parent topic: [Locate actions](#)

Related reference:

[FindNextKeyList](#)
[FindKeyList](#)
[AddKeyList](#)

FindLastKeyList_InZone

Locates the last occurrence of a word or phrase that matches one of the current field.

Syntax

```
bool FindLastKeyList_InZone (StrParam)
```

Parameters

The name of the Keyword text file. The file will contain a list of words or phrases, separated by new lines, that will be used for matching.

The file name can be provided in one of two ways:

1. A full path name of the file, including the .key extension.
2. The file name only, with no extension specified. The application will look in the process directory and the file must have a .key extension.

Returns

True if at least one word on the page matches any word or pattern in the Keyword file. Otherwise, False.

Level

Field level.

Details

Opens the Keyword file you specify as a parameter, then checks the words or phrases in the current bound zone against the keywords in the list to find a match. The location of the found word or phrase will be remembered so the result can be used by subsequent actions. Word matching is case sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes "will" and the recognition read "wi11", a match will still occur.

Common substitutions include characters: B8, Z2, S5, oO0 and iItl1.

Example

```
FindLastKeyList_InZone ("ClaimsData")
```

This action searches the current bound zone for the last instance of a match with the first keyword in the Keyword file (ClaimsData.key). If successful, the search stops and remembers the location of that word for subsequent actions; if not, the action continues searching using the next word in ClaimsData.key, matching the last occurrence of the word, repeating this search pattern until a match is found, or until there are no more keywords.

Parent topic: [Locate actions](#)

Related reference:
[FindKeyList_InZone](#)

FindLastRegEx

Same as the FindLastWord action, except that it supports regular expressions.

Syntax

```
bool FindLastRegEx (StrParam)
```

Parameters

A word or phrase to find on the current page. The parameter is expected to be a Regular Expression.

Returns

True if the word or phrase is located on the page. Otherwise, False.

Level

Page level.

Details

Locates the last occurrence of a word or phrase on the current page where the input search term is specified as a regular expression. The search is started from the first word of the current page. The location of the found word or phrase will be remembered so the result can be used by subsequent actions. The search is case sensitive.

Example

```
FindLastRegEx ("Da?e")  
GoRightWord ("1")  
IsDateValue ()  
UpdateField ()
```

FindLastRegEx will look for the last occurrence of a word that matches the regular expression. If it match is found, it will continue processing by moving one word right to the result of RegExFind and act on that word. In this example, if it is a date format, UpdateField will place the value into the current field.

Parent topic: [Locate actions](#)

Related reference:

[FindRegExList](#)

[RegExFind_InZone](#)

[RegExFindNext](#)

FindLastRegEx_InZone

Same as the FindLastWord_InZone action, except that it supports regular expressions.

Syntax

```
bool FindLastRegEx_InZone (StrParam)
```

Parameters

A word or phrase to find in the current field. The parameter is expected to be a Regular Expression.

Returns

True if the word or phrase is located in the field. Otherwise, False.

Level

Field level.

Details

Locates the last occurrence of a word or phrase in the current field where the input search term is specified as a regular expression. The search is started from the first word of the current field. The location of the found word or phrase will be remembered so the result can be used by subsequent actions. The search is case sensitive.

Example

```
FindLastRegEx_InZone ("Da?e")
GoRightWord ("1")
IsDateValue ()
UpdateField ()
```

FindLastRegEx will look for the last occurrence of a word that matches the regular expression. If it match is found, it will continue processing by moving one word right to the result of RegExFind and act on that word. In this example, if it is a date format, UpdateField will place the value into the current field.

Parent topic: [Locate actions](#)

Related reference:

[FindRegExList](#)

[RegExFind_InZone](#)

[RegExFindNext](#)

FindLastRegExList

Same as the FindLastKeyList action, except that it supports regular expressions.

Syntax

```
bool FindLastRegExList (StrParam)
```

Parameters

The name of the Keyword text file. The file will contain a list of words or phrases, separated by new lines, that will be used for matching. The entries in the keyword file are expected to be regular expressions.

The file name can be provided in one of two ways:

1. A full path name of the file, including the .key extension.

2. The file name only, with no extension specified. The application will look in the process directory and the file must have a .key extension.

Returns

True if at least one word on the page matches any word or pattern in the Keyword file. Otherwise, False.

Level

Page level.

Details

Opens the Keyword file you specify as a parameter, then checks the words or phrases on the current page against the keywords in the list to find a match. The entries in the keyword file are expected to be regular expressions. To perform the action, the search first looks for the last occurrence of the first word or phrase in the keyword file. If the word is found on the page, the search stops at the last occurrence of the word. If no match is found, the next line from the keyword file is read and again the entire page is searched. This process continues until a match is found or all of the lines in the keyword file have been read. The location of the last occurrence of a found word or phrase will be remembered so the result can be used by subsequent actions. Word matching is case sensitive.

Example

```
FindLastRegExList("InvoiceNum")
```

This action searches the current page, from the first word of the current page, for the last occurrence of the first keyword in the Keyword file (InvoiceNum.key). If successful, the search stops and remembers the location of that word for subsequent actions; if not, the action continues searching for the last occurrence of the next word in InvoiceNum.key, starting from starting of the current page, repeating this search pattern until a match is found, or until there are no more keywords.

Parent topic: [Locate actions](#)

Related reference:

[FindRegExList_InZone](#)

[FindLastKeyList](#)

[FindNextRegExList](#)

[AddKeyList](#)

FindLastRegExList_InZone

Same as the FindLastRegExList action, except that it searches the current field only.

Syntax

```
bool FindLastRegExList_InZone (StrParam)
```

Parameters

The name of the Keyword text file. The file will contain a list of words or phrases, separated by new lines, that will be used for matching. The entries in the keyword file are expected to be regular expressions.

The file name can be provided in one of two ways:

1. A full path name of the file, including the .key extension.

2. The file name only, with no extension specified. The application will look in the process directory and the file must have a .key extension.

Returns

True if at least one word in the field matches any word or pattern in the Keyword file. Otherwise, False.

Level

Field level.

Details

Opens the Keyword file you specify as a parameter, then checks the words or phrases in the current field against the keywords in the list to find a match. The entries in the keyword file are expected to be regular expressions.

To perform the action, the search first looks for the last occurrence of the first word or phrase in the keyword file. If the word is found on the page, the search stops at the last occurrence of the word. If no match is found, the next line from the keyword file is read and again the entire field is searched. This process continues until a match is found or all of the lines in the keyword file have been read. The location of the last occurrence of a found word or phrase will be remembered so the result can be used by subsequent actions. Word matching is case sensitive.

Example

```
FindLastRegExList_InZone ("InvoiceNum")
```

This action searches the current page, from the first word of the current page, for the last occurrence of the first keyword in the Keyword file (InvoiceNum.key). If successful, the search stops and remembers the location of that word for subsequent actions; if not, the action continues searching for the last occurrence of the next word in InvoiceNum.key, from starting of the current page, repeating this search pattern until a match is found, or until there are no more keywords.

Parent topic: [Locate actions](#)

Related reference:

[FindRegExList_InZone](#)

[FindLastKeyList](#)

[FindNextRegExList](#)

[AddKeyList](#)

FindLastWord

Locates the last occurrence of the specified word or phrase on the current page.

Syntax

```
bool FindLastWord (StrParam)
```

Parameters

A word or phrase to find on the page.

Returns

True, if the word or phrase is on the page. Otherwise, False.

Level

Page level.

Details

The current page is searched to find the last occurrence of a word or phrase. The location of the last word or phrase that matches the parameter is remembered and it can be used by subsequent actions.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes `will` and the recognition that is read is `wi11`, a match still occurs.

Common substitutions include characters: B8, Z2, S5, o00, and iIt1

Example

```
FindLastWord("Total")
GoRightWord("1")
IsCurrency()
```

In this example, the action finds the last instance of `Total` on the current page. Then, it moves right one word and checks to be sure that the word has a `Currency` value.

Parent topic: [Locate actions](#)

Related reference:

[WordFind](#)

[WordFindNext](#)

FindLastWord_InZone

Locates the last occurrence of the specified word or phrase on the current field.

Syntax

```
bool FindLastWord_InZone (StrParam)
```

Parameters

The word or phrase to locate within the current zone.

Returns

True if the word or phrase is located in the field. Otherwise, False.

Level

Page level.

Details

The current field will be searched to find the word or phrase. The location of the last occurrence of the word or phrase that matches the parameter will be remembered so it can be utilized by subsequent actions. Word matching is case sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes "will" and the recognition read "wi11", a match will still occur.

Common substitutions include characters: B8, Z2, S5, oO0 and iItl1.

Example

```
FindLastWord_InZone ("Total")
GoRightWord ("1")
IsCurrency ()
```

This sequence attempts to find the last occurrence of "Total" in the current zone. It then locates and validates a Currency amount for the Total field, assuming the currency amount is to the right of the word "Total".

Parent topic: [Locate actions](#)

Related reference:

[WordFind_InZone](#)

[WordFindNext_InZone](#)

FindNextDBList

Same as the FindDBList action, except that it locates the next instance.

Syntax

```
bool FindNextDBList (StrParam)
```

Parameters

An SQL statement whose result returns a word or phrase, or a list of word or phrases.

Returns

True, if the word or phrase is on the page. Otherwise, False.

Level

Page level.

Details

This action generates an SQL query against the Lookup database, returning a list of words or phrases that are on the current page. Starting from the location of a previously found word, the search is performed. Initially, the first listed word or phrase from the SQL query is searched for on the page. If there is no match, it will search for the next word or phrase that is returned from the database. The search continues until a match is found or none of the returned words are contained on the page. The location of the found word or phrase is remembered so the result can be used by subsequent actions. The search is case-sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes `will` and the recognition that is read is `wi11`, a match still

occurs.

Common substitutions include characters: B8, Z2, S5, o00, and iItl1.

Example

```
WordFind("Hello")
FindNextDBList("Select LastName from Providers")
```

In this example, when the first word `Hello` is found, the `LastName` list is obtained from a provider table. Then, the first matching occurrence on the page that occurs after the word `Hello` is found.

Parent topic: [Locate actions](#)

Related reference:

[FindDBList](#)

[FindDBList_InZone](#)

FindNextDBList_InZone

Same as the `FindNextDBList` action, except that it searches the current field only.

Syntax

```
bool FindNextDBList_InZone (StrParam)
```

Parameters

An SQL statement whose result returns a word or phrase, or a list of word or phrases.

Returns

True, if the word or phrase is in the field. Otherwise, False.

Level

Field level.

Details

This action generates an SQL query against the lookup database, returning a list of words or phrases that are in the current field. Starting from the location of a previously found word, the search is performed. Initially, the first listed word or phrase from the SQL query is searched for on the page. If there is no match, it will search for the next word or phrase that is returned from the database. The search continues until a match is found or none of the returned words are contained on the page. The location of the found word or phrase is remembered so the result can be used by subsequent actions. The search is case-sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes `will` and the recognition that is read is `wi11`, a match still occurs.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions.

Common substitutions include characters: B8, Z2, S5, o00, and iItl1

Example

```
WordFind_InZone("Hello")
FindNextDBList("Select LastName from Providers")
```

In this example, when the first word `Hello` is found in the current field, the `LastName` list is obtained from a provider table. Then, the first matching occurrence in the field that occurs after the word `Hello` is found.

Parent topic: [Locate actions](#)

Related reference:

[WordFind_InZone](#)

[FindDBList](#)

FindNextKeyList

Same as the `FindKeyList` action, except that it locates the next instance.

Syntax

```
bool FindNextKeyList (StrParam)
```

Parameters

The name of the Keyword text file. The file will contain a list of words or phrases, separated by new lines, that will be used for matching.

The file name can be provided in one of two ways:

1. A full path name of the file, including the `.key` extension.
2. The file name only, with no extension specified. The application will look in the process directory and the file must have a `.key` extension.

Returns

True if at least one word or phrase on the page, starting from the previously located word, matches any word or pattern in the Keyword file. Otherwise, False.

Level

Field level.

Details

Opens the Keyword file you specify as a parameter, then checks the words or phrases on the current page against the keywords in the list to find a match. The search starts from the last word that had been found using an action such as `FindKeyList` or `FindNextKeyList`. The location of the found word or phrase that matches an entry in the keyword file will be remembered so the result can be used by subsequent actions. Matching is case sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes "will" and the recognition read "wi11", a match will still occur.

Common substitutions include characters: B8, Z2, S5, oO0 and iItl1.

Example

```
FindKeyList("Tax")  
FindNextKeyList("IRS")
```

Starting from the word or phrase that was located in the FindKeyList action, this action searches the current page, starting from that previously found word, for the first word or phrase in the first line of the Keyword file (IRS.key). If successful, the search stops and remembers the location of that new word for subsequent actions; if not, the action continues searching using the next word or phrase in IRS.key, starting again from the location of the word previously found by FindKeyList(Tax), repeating this search pattern until a match is found, or until there are no more keywords in the file.

Parent topic: [Locate actions](#)

Related reference:

[FindLastKeyList](#)

[AddKeyList](#)

[FindKeyList_InZone](#)

[FindNextKeyList_InZone](#)

FindNextKeyList_InZone

Same as the FindNextKeyList action, except that it searches the current field only.

Syntax

```
bool FindNextKeyList_InZone (StrParam)
```

Parameters

The name of the Keyword text file. The file will contain a list of words or phrases, separated by new lines, that will be used for matching.

The file name can be provided in one of two ways:

1. A full path name of the file, including the .key extension.
2. The file name only, with no extension specified. The application will look in the process directory and the file must have a .key extension.

Returns

False if the Fingerprint file (.cco) of the current source page has not been loaded, or is empty. Otherwise, True.

Level

Field level.

Details

Opens the Keyword file you specify as a parameter, then checks the words or phrases in the current field against the keywords in the list to find a match. The search starts from the last word that had been found using an action such as FindKeyList_InZone or FindNextKeyList_InZone. The location of the found word or phrase that matches an entry in the keyword file will be remembered so the result can be used by subsequent actions. Matching is case sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes "will" and the recognition read "wi11", a match will still occur.

Common substitutions include characters: B8, Z2, S5, o00 and iIt1.

Example

```
FindKeyList_InZone("Inventory")
FindNextKeyList_InZone("Parts")
```

This action searches the current field, starting from the first word of the zone, for the first occurrence of a word or phrase in the Keyword file "Inventory.key". If a match is found, the application will then look for a word from within the Keyword file "Parts.key", starting from the previously found word from the FindKeyList_InZone action and using the first word or phrase in the "Parts.key" file. If successful, the search stops and remembers the location of the new word for subsequent actions; if not, the action continues searching using the next word in "Parts.key", starting again from the location of previously found word in the zone, repeating this search pattern until a match is found, or until there are no more keywords in the file.

Parent topic: [Locate actions](#)

FindNextRegExList

Same as the FindRegExList action, except that it locates the next instance.

Syntax

```
bool FindNextRegExList (StrParam)
```

Parameters

The name of the Keyword text file. The file will contain a list of words or phrases, separated by new lines, that will be used for matching.

The file name can be provided in one of two ways:

1. A full path name of the file, including the .key extension.
2. The file name only, with no extension specified. The application will look in the process directory and the file must have a .key extension.

Returns

True if at least one word on the page, that follows the result of a previous find, matches any word or pattern in the Keyword file. Otherwise, False.

Level

Field level only.

Details

Opens the Keyword file you specify as a parameter, starting from the location of a previous find, this action checks the words or phrases on the current page against the keywords in the list to find a match. The entries in the keyword file are expected to be regular expressions.

To perform the action, the search first looks for the first word or phrase in the keyword file. Starting from the location of the last find, if the word is found, the search stops. If no match is found, the next line from the keyword file is read and again the search starts from the result of a previous find. This process continues until a match is found or all of the lines in the keyword file have been read. The location of the found word or phrase that matches an entry in the keyword file will be remembered so the result can be used by subsequent actions. Word matching is case sensitive.

Example

```
FindRegExList ("InvoiceNum")  
FindNextRegExList ("InvoiceNum")
```

This action searches the current page, starting from the result of the FindRegExList action, for the first keyword in the Keyword file (InvoiceNum.key). If successful, the search stops and remembers the location of that word for subsequent actions; if not, the action continues searching for the next word in InvoiceNum.key, repeating this search pattern until a match is found, or until there are no more keywords. Although this example shows FindNextRegExList using the same keyword file, it is allowed to use a different keyword file.

Parent topic: [Locate actions](#)

Related reference:

[FindRegExList_InZone](#)

[FindLastKeyList](#)

[FindNextRegExList_InZone](#)

[AddKeyList](#)

FindNextRegExList_InZone

Same as the FindNextRegExList action, except that it searches the current field only.

Syntax

```
bool FindNextRegExList_InZone (StrParam)
```

Parameters

The name of the Keyword text file. The file contains a list of words or phrases, separated by new lines, that will be used for matching. The file name can be provided in one of two ways:

1. A full path name of the file, including the .key extension.
2. The file name only, with no extension specified. The application looks in the process directory and the file must have a .key extension.

Returns

True if at least one word in the field, that follows the result of a previous find, matches any word or pattern in the Keyword file. Otherwise, False.

Level

Field level only.

Details

Opens the Keyword file you specify as a parameter, starting from the location of a previous find, this action checks the words or phrases in the current field against the keywords in the list to find a match. The entries in the keyword file are expected to be regular expressions.

To perform the action, the search first looks for the first word or phrase in the keyword file. Starting from the location of the last find, if the word is found, the search stops. If no match is found, the next line from the keyword file is read and again the search starts from the result of a previous find.

This process continues until a match is found or all of the lines in the keyword file have been read. The location of the found word or phrase that matches an entry in the keyword file is remembered so the result can be used by subsequent actions. Word matching is case sensitive.

Example

```
FindRegExList_InZone("InvoiceNum")
FindNextRegExList_InZone("InvoiceNum")
```

This action searches the current page, starting from the result of the FindRegExList_InZone action, for the first keyword in the Keyword file (InvoiceNum.key). If successful, the search stops and remembers the location of that word for subsequent actions; if not, the action continues searching for the next word in InvoiceNum.key, repeating this search pattern until a match is found, or until there are no more keywords. Although this example shows FindNextRegExList_InZone using the same keyword file, it is allowed to use a different keyword file.

Parent topic: [Locate actions](#)

Related reference:

[FindRegExList_InZone](#)

[FindLastKeyList](#)

[FindNextRegExList_InZone](#)

[AddKeyList](#)

FindRegExList

Same as the FindKeyList action, except that it supports regular expressions.

Syntax

```
bool FindRegExList (StrParam)
```

Parameters

The name of the Keyword text file. The file will contain a list of words or phrases, separated by new lines, that will be used for matching.

The file name can be provided in one of two ways:

1. A full path name of the file, including the .key extension.
2. The file name only, with no extension specified. The application will look in the process directory and the file must have a .key extension.

Returns

True if at least one word on the page matches any word or pattern in the Keyword file. Otherwise, False.

Level

Field level only.

Details

Opens the Keyword file you specify as a parameter, then checks the words or phrases on the current page against the keywords in the list to find a match. The entries in the keyword file are expected to be regular expressions.

To perform the action, the search first looks for the first word or phrase in the keyword file. If the word is found on the page, the search stops. If no match is found, the next line from the keyword file is read and again the entire page is searched. This process continues until a match is found or all of the lines in the keyword file have been read. The location of the found word or phrase will be remembered so the result can be used by subsequent actions. Word matching is case sensitive.

Example

```
FindRegExList ("InvoiceNum")
```

This action searches the current page, from the first word of the current page, for the first keyword in the Keyword file (InvoiceNum.key). If successful, the search stops and remembers the location of that word for subsequent actions; if not, the action continues searching for the next word in InvoiceNum.key, starting from first word of the current page, repeating this search pattern until a match is found, or until there are no more keywords.

Parent topic: [Locate actions](#)

Related reference:

[FindRegExList_InZone](#)

[FindLastKeyList](#)

[FindNextRegExList](#)

FindRegExList_InZone

Same as the FindRegExList action, except that it searches the current field only.

Syntax

```
bool FindRegExList_InZone (StrParam)
```

Parameters

The name of the Keyword text file. The file will contain a list of words or phrases, separated by new lines, that will be used for matching.

The file name can be provided in one of two ways:

1. A full path name of the file, including the .key extension.
2. The file name only, with no extension specified. The application will look in the process directory and the file must have a .key extension.

Returns

True if at least one word on the page matches any word or pattern in the Keyword file. Otherwise, False.

Level

Field level only.

Details

Opens the Keyword file you specify as a parameter, then checks the words or phrases on the current page against the keywords in the list to find a match. The entries in the keyword file are expected to be regular expressions.

To perform the action, the search first looks for the first word or phrase in the keyword file. If the word or phrase is found in the current field zone, the search stops. If no match is found, the next line from the keyword file is read and again the current field zone is searched. This process continues until a match is found or all of the lines in the keyword file have been read. The location of the found word or phrase that matches an entry in the keyword file will be remembered so the result can be used by subsequent actions. Word matching is case sensitive.

Example

```
FindRegExList_InZone ("InvoiceNum")
```

This action searches the current page, from the first word of the current field, for the first keyword in the Keyword file (InvoiceNum.key). If successful, the search stops and remembers the location of that word for subsequent actions; if not, the action continues searching for the next word in InvoiceNum.key, starting from first word of the current field, repeating this search pattern until a match is found, or until there are no more keywords.

Parent topic: [Locate actions](#)

Related reference:

[FindRegExList](#)

[FindLastKeyList](#)

[FindNextRegExList](#)

[AddKeyList](#)

GetSelectedBlockType

Obtains the type of the currently selected block and stores it in the DCO location indicated by the target parameter. This action uses the document block layout that is created by the DocumentAnalytics actions.

Important: This action is currently released as a preview. If you use this action in your application, you might need to make updates to your application if a new version of this action released.

Syntax

```
bool GetSelectedBlockType (string varTarget)
```

Parameters

string varTarget

Target

A smart parameter that defines the target variable to store the type of the currently selected block.

Returns

True if the type was successfully stored, otherwise False.

Level

Page or field.

Details

Obtains the type of the currently selected block and stores it in the DCO location indicated by the target parameter. This action uses the document block layout created by the DocumentAnalytics actions.

Example

```
RegExFind_InBlock("Da.e")
GetSelectedBlockType("@X.SelectedBlock")
```

Finds the block that contains the matching expression and then stores the name of the currently selected block in a variable called `SelectedBlock` on the current DCO object.

If the variable does not exist, it will be created. In this example, the name of the stored block would be `Word` because the currently selected block is the found word block.

Parent topic: [Locate actions](#)

Related reference:

[DocumentAnalytics actions](#)

GoAboveWord

Moves up the specified number of lines from the previously found word or phrase.

Syntax

```
bool GoAboveWord (strParam)
```

Parameters

An integer indicating the number of lines to move up.

Returns

True if a word is found. Otherwise, False.

Level

Page or field level.

Details

Starting from a word or phrase found by a previous Locate action, the location position is moved to a word directly above it by the number of lines indicated by the input parameter. Subsequent Locate actions can then perform additional operations on this located word. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Attention: If the action is added with an empty or non-numeric parameter, the action will default the argument to '1'.

Example

```
WordFind("Total")
GoAboveWord("1")
GoRightWord("1")
UpdateField()
```

This sequence finds the word which is the Tax field's entered value ("29.78") when a fingerprint has these fields and values:

```
Tax 29.78
Total 234.70
```

Parent topic: [Locate actions](#)

Related reference:

[GoBelowWord](#)

GoBelowWord

Moves down the specified number of lines from the previously found word or phrase.

Syntax

```
bool GoBelowWord (strParam)
```

Parameters

An integer indicating the number of lines to move down.

Returns

True if the word or phrase is found. Otherwise, False.

Level

Page or field level.

Details

Starting from a word or phrase found by a previous Locate action, the location position is moved to a word directly below it by the number of lines indicated by the input parameter. Subsequent Locate actions can then perform additional operations on this located word. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Attention: If the action is added with an empty or non-numeric parameter, the action will default the argument to '1'. This action is always called after a search action has been used.

Example

```
WordFind("Tax")
GoBelowWord("1")
GoRightWord("1")
UpdateField()
```

This sequence finds the word which is the Total field's entered value ("234.70") when a fingerprint has these fields:

```
Tax 29.78
Total 234.70
```

Parent topic: [Locate actions](#)

Related reference:

[GoAboveWord](#)

[WordFind](#)

[FindKeyList](#)

[RegExFind](#)

GoDownLine

Moves down the specified number of lines from the previously found word or phrase and selects the first word.

Syntax

```
bool GoDownLine (strParam)
```

Parameters

An integer indicating the number of lines to move down below the current line.

Returns

True if a line is found. Otherwise, False.

Level

Page or field level.

Details

Starting from the current position, the location position is moved down by the number of lines indicated in the input parameter and is moved to the first word of that line.

Attention: If the action is added with an empty or non-numeric parameter, the action will default the argument to '1'. This action is always called after a search action has been used.

Example

```
FindKeyList("Invoice")
GoDownLine("1")
GoRightWord("1")
UpdateField()
```

This sequence finds the word which is the Number field's entered value ("10034") when a fingerprint has these fields:

```
INVOICE Number: 10034
```

Parent topic: [Locate actions](#)

Related reference:

[GoUpLine](#)

[WordFind](#)

[FindKeyList](#)

GoFirstLine

Moves to the first line of the page when you are running full page recognition.

Syntax

```
bool GoFirstLine ()
```

Parameters

None.

Returns

False if the page's Fingerprint file(.cco) is unavailable or empty. Otherwise, True.

Level

Page or field level.

Details

Starting from the current position, the location position is moved to the first line of the current zone, or to the first line on the page if a zone is not present. Subsequent Locate actions can then perform additional operations from this position. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
GoFirstLine ()
```

Parent topic: [Locate actions](#)

Related reference:

[WordFind](#)

[FindKeyList](#)

[RegExFind](#)

GoFirstWord

Moves to the first word on the current line.

Syntax

```
bool GoFirstWord ()
```

Parameters

None.

Returns

False if the page's Fingerprint file (.cco) is not available or if it is empty. Otherwise, True.

Level

Page or field level.

Details

Starting from a word or phrase found by a previous Locate action, the location position is moved to the first word on the current line. If the page's Fingerprint file (.cco) does not have a line position, the action defaults to the first word on the first line of the current zone, or on the first line of the current page if a zone is not present. Subsequent Locate actions can then perform additional operations on this located word. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
GoFirstWord()
```

Parent topic: [Locate actions](#)

Related reference:

[WordFind](#)

[FindKeyList](#)

[RegExFind](#)

GoLastLine

Moves to the last line of the page when you are running full page recognition.

Syntax

```
bool GoLastLine ()
```

Parameters

None.

Returns

False if the source page's Fingerprint file (.cco) is not available, or if it is empty. Otherwise, True.

Level

Page or field.

Details

Starting from the current position, the location position is moved to the last line and word of the current zone, or to the last line and word on the page if a zone is not present. Subsequent Locate actions can then perform additional operations from this position. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
GoLastLine()
```

Parent topic: [Locate actions](#)

Related reference:

[WordFind](#)

GoLastWord

Moves to the last word on the current line.

Syntax

```
bool GoLastWord ()
```

Parameters

None.

Returns

False if the Fingerprint file (.cco) is not available, or if it is empty. Otherwise, True.

Level

Page or field level.

Details

Starting from a word or phrase found by a previous Locate action, the location position is moved to the last word on the current line. If the page's Fingerprint file (.cco) does not have a line position, the action defaults to the last word on the first line of the current zone, or on the first line of the current page if a zone is not present. Subsequent Locate actions can then perform additional operations on this located word. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
GoLastWord ()
```

Parent topic: [Locate actions](#)

Related reference:

[WordFind](#)
[FindKeyList](#)
[RegExFind](#)

GoLeftWord

Moves the specified number of words to the left of the previously found word or phrase.

Syntax

```
bool GoLeftWord (strParam)
```

Parameters

An integer indicating the number of words to move to the left.

Returns

True if a word is found. Otherwise, False.

Level

Page or field level.

Details

Starting from a word or phrase found by a previous Locate action, the location position is moved to the left on the current line by the number of words indicated by the input parameter. Subsequent Locate actions can then perform additional operations on this located word. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Attention: If the action is added with an empty or non-numeric parameter, the action will default the argument to '1'. This action is always called after a search action has been used.

Example

```
WordFind("Total")
GoLeftWord("1")
GoBelowWord("1")
UpdateField()
```

If the fingerprint has a table with columns such as Total and Tax, the actions above will locate the Tax amount below ("344.76"):

```
Tax Total
344.76 13,774.00
```

Parent topic: [Locate actions](#)

Related reference:

[GoRightWord](#)

[WordFind](#)

[FindKeyList](#)

GoRightWord

Moves the specified number of words to the right of the previously found word or phrase.

Syntax

```
bool GoRightWord (strParam)
```

Parameters

An integer indicating the number of words to move to the right.

Returns

True if a word is found. Otherwise, False.

Level

Page or field level.

Details

Starting from a word or phrase found by a previous Locate action, the location position is moved to the right on the current line by the number of words indicated by the input parameter. Subsequent Locate actions can then perform additional operations on this located word. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Attention: If the action is added with an empty or non-numeric parameter, the action will default the argument to '1'. This action is always called after a search action has been used.

Example

```
WordFind("Tax") GoRightWord("1")
GoBelowWord("1")
UpdateField()
```

If the fingerprint has a table with columns such as Total and Tax, the actions above will locate the Total amount below ("13,774.00"):

```
Tax Total 344.76 13,774.00
```

Parent topic: [Locate actions](#)

GoSiblingBlockNext

The currently selected block in the document layout is changed to the next sibling block at the same level in the structure. The number of siblings that are moved is determined by the input parameter. This action uses the document block layout that is created by the DocumentAnalytics actions.

Important: This action is currently released as a preview. If you use this action in your application, you might need to make updates to your application if a new version of this action released.

Syntax

```
bool GoSiblingBlockNext (string count)
```

Parameters

string count

count

The number of sibling blocks to move. Smart parameters are supported.

Returns

True if the sibling is found at the location that is specified from the currently selected block. Otherwise, False.

Level

Page or field.

Details

When a sibling block is selected, the currently selected block becomes the sibling block and the currently selected word is changed to the first word in the new block and that word might be in a nested block. Any searches that look for the next instance will start from the location of this selected word.

For example, if the currently selected block is a line and the parameter is two, then the currently selected block moves down two lines in the structure and the currently selected block will be the sibling line and the selected word will be the first word in the new line. If there is no sibling at that location, then the current block is not changed and the action returns false.

Example

```
RegexFind_InBlock("Da.e")
SelectParentBlockType("Block")
GoSiblingBlockNext("2")
RegexFind_InBlock("Qty")
```

This example finds the matching text for the regular expression `Da.e`, then select the parent block of type `Block`. It then moves down two blocks of type `Block`. Now, the currently selected block is a type block and the currently selected word is the first word in that block. Then, another search is run within that new block for `Qty`. If `Qty` does not exist within the bounds of that new block, the action returns false.

Parent topic: [Locate actions](#)

Related reference:

[GoSiblingBlockPrevious](#)

GoSiblingBlockPrevious

The currently selected block in the document layout is changed to the previous sibling block at the same level in the structure. The number of siblings moved is determined by the input parameter. This action uses the document block layout that is created by the DocumentAnalytics actions.

Important: This action is currently released as a preview. If you use this action in your application, you might need to make updates to your application if a new version of this action released.

Syntax

```
bool GoSiblingBlockPrevious (string count)
```

Parameters

string count

count

The number of sibling blocks to move. Smart parameters are supported.

Returns

True if the sibling is found at the location specified from the currently selected block. Otherwise, False.

Level

Page or field.

Details

When a sibling block is selected, the currently selected block becomes the sibling block and the currently selected word is changed to the first word in the new block, and that word might be in a nested block. Any searches that look for the next instance will start from the location of this selected word.

For example, if the currently selected block is a line and the parameter is two, then the currently selected block will move up two lines in the structure. If there is no sibling at that location, then the current block is not changed and the action returns false.

Example

```
RegExFind_InBlock("Da.e")
SelectParentBlockType("Block")
GoSiblingBlockPrevious("2")
RegExFind_InBlock("Qty")
```

This example will find the matching text for the regular expression `Da.e`, then select the parent block of type `Block`. The example then moves up two blocks of type `Block`. Now, the currently selected block is a type block and the currently selected word is the first word in that block. Then another search is performed within that new block for `Qty`. If `Qty` does not exist within the bounds of that new block, the action will return false.

Parent topic: [Locate actions](#)

GoUpLine

Moves up the specified number of lines from the previously found word or phrase and selects the first word.

Syntax

```
bool GoUpLine (strParam)
```

Parameters

An integer indicating the number of lines to move up above the current line.

Returns

True if a line is found. Otherwise, False.

Level

Page or field level.

Details

Starting from the current position, the location position is moved up by the number of lines indicated in the input parameter and is moved to the first word of that line.

Attention: If the action is added with an empty or non-numeric parameter, the action will default the argument to '1'. This action is always called after a search action has been used.

Example

```
FindKeyList("Invoice")
GoUpLine("1")
```

```
GoRightWord("1")
UPdateField()
```

This sequence finds and extracts the entered value of the Date field ("2/13/03"), in a fingerprint with these words:

```
Date: 2/13/03
INVOICE
```

Parent topic: [Locate actions](#)

Related reference:

[GoDownLine](#)

[WordFind](#)

[FindKeyList](#)

[RegExFind](#)

GroupWords

Groups words to the left and right of the previously found word if they are no more than the specified number of character widths apart.

Syntax

```
bool GroupWords (nSpaces)
```

Parameters

Long value of the maximum character width separating words to the right and left of the current word.

Returns

Always True.

Level

Page or field level.

Details

Groups any words to the left and right of a located word if the target words are themselves separated by a character width equal to or less than the character width you specify as a parameter. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
WordFind("Treasury")
GroupWords("1.5")
```

```
If a line contains these words:
20 000 U S Treasury 7 33
```

the GroupWords action merges "20" with "000"; "U" with "S" and "7" with "33" to produce:

```
20 000 U S Treasury 7 33
```

Parent topic: [Locate actions](#)

Related reference:

[GroupWordsLEFT](#)

[GroupWordsRIGHT](#)

GroupWordsLEFT

Groups words to the left of the previously found word if they are no more than the specified number of character widths apart.

Syntax

```
bool GroupWordsLEFT (nSpaces)
```

Parameters

Long value of the maximum character width separating words to the left of the current word.

Returns

Always True.

Level

Page or field level.

Details

Groups only words to the left of the located word if the target words are separated by a character width equal to or less than the character width you specify as a parameter. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
WordFind("000")  
GroupWordsLeft("2")
```

```
If a line contains these words:  
Found 20 000 US Treasury bills.
```

the GroupWordsLeft action merges "20" with "000" to produce:

```
20 000
```

Parent topic: [Locate actions](#)

Related reference:

[GroupWords](#)

[GroupWordsRIGHT](#)

GroupWordsRIGHT

Groups words to the right of the previously found word if they are no more than the specified number of character widths apart.

Syntax

```
bool GroupWordsRIGHT (nSpaces)
```

Parameters

Long value of the maximum character width separating words to the right of the current word.

Returns

Always True.

Level

Page or field level.

Details

Groups words to the right of the located word if the target words are separated by a character width equal to or less than the character width you specify as a parameter. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
WordFind("US")
GroupWordsRight("2")
```

```
If a line contains these words:
Found 20 000 US Treasury bills.
```

the GroupWordsRight action merges "US" with "Treasury" to produce:

```
US Treasury
```

Parent topic: [Locate actions](#)

Related reference:

[GroupWords](#)

[GroupWordsLEFT](#)

IsAlpha

Determines whether the specified percentage of characters in a located word is letters (defaults to 100%)

Syntax

```
bool IsAlpha (StrParam)
```

Parameters

An integer (0-100) indicating the minimum percentage of characters that must be alphabetic. If no value is provided, the percentage defaults to 100; all characters must be alphabetic.

Returns

True if the minimum percentage of characters specified by the parameter is alphabetic. Otherwise, False.

Level

Page or field level.

Details

Using the current location of a previously located word or phrase, this action determines if the characters are alphabetic. By testing the type of characters recognized in the current word or phrase, it is possible for an application to determine it has located the type of data that is required, and then take subsequent actions based on the result of the test. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
FindKey (Name)  
GoRightWord ("1")  
IsAlpha ("100")
```

If the located word's recognized value is: ABC1

```
IsAlpha ("75") returns True
```

```
IsAlpha ("80") returns False
```

Parent topic: [Locate actions](#)

Related reference:

[IsCurrency](#)

[IsDateValue](#)

[IsNumber](#)

IsCurrency

Determines whether the value of the located word is a currency value. The value contains numbers and includes a two-digit decimal amount.

Syntax

```
bool IsCurrency ()
```

Parameters

None.

Returns

True if the located value is currency. Otherwise, False.

Level

Page or field level.

Details

Using the current location of a previously located word or phrase, this action determines if the characters are in a valid currency format and are a valid currency value. By testing the type of characters recognized in the current word or phrase, it is possible for an application to determine it has located the type of data that is required, and then take subsequent actions based on the result of the test. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Attention: This action is NOT dynamic locale aware and uses a simple regex and separator character test.

Example

```
If the recognized value of the word or phrase is: 12.00  
IsCurrency() returns TRUE
```

```
If the recognized value of the word or phrase is: 12,00  
IsCurrency() returns TRUE
```

```
If the recognized value of the word or phrase is: 1200  
IsCurrency() returns FALSE
```

```
If the recognized value of the word or phrase is: 1,200.00  
IsCurrency() returns TRUE
```

```
If the recognized value of the word or phrase is: $12.00  
IsCurrency() returns TRUE
```

Parent topic: [Locate actions](#)

Related reference:

[IsAlpha](#)

[IsDateValue](#)

[IsNumber](#)

IsDateValue

Determines whether the value of the located word is in one of the supported date formats.

Syntax

```
bool IsDateValue ()
```

Parameters

None.

Returns

True if the located value is an acceptable Date. Otherwise, False.

Level

Page or field level.

Details

Using the current location of a previously located word or phrase, this action determines if the characters are in a valid date format and are a valid date. By testing the type of characters recognized in the current word or phrase, it is possible for an application to determine it has located the type of data that is required, and then

take subsequent actions based on the result of the test. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Attention: This action is dynamic locale aware.

Example

```
If the located word's recognized value is: 01/01/2003  
IsDateValue() returns TRUE
```

```
If the located word's recognized value is: January 01,2003  
IsDateValue() returns TRUE
```

```
If the located word's recognized value is: 13/13/2003  
IsDateValue() returns FALSE
```

Parent topic: [Locate actions](#)

Related reference:

[IsAlpha](#)

[IsCurrency](#)

[IsNumber](#)

[WordFind](#)

[FindKeyList](#)

[RegExFind](#)

IsNumber

Determines whether the specified percentage of characters in a located word is numbers (defaults to 100%)

Syntax

```
bool IsNumber (StrParam)
```

Parameters

An integer (0-100) indicating the minimum percentage of characters that must be numeric. If no parameter is specified, the value defaults to 100 percent; all characters must be numeric.

Returns

True if the located value meets the parameter's requirement for an integer. Otherwise, False.

Level

Page or field level.

Details

Using the current location of a previously located word or phrase, this action determines if the characters are numeric. By testing the type of characters recognized in the current word or phrase, it is possible for an application to determine it has located the type of data that is required, and then take subsequent actions based on the result of the test. This action does not consider the decimal symbol, digit grouping symbol or a currency symbol to be numeric. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
WordFind("Total")
GoRightWord("1")
IsNumber("100")
```

If the located word's recognized value is: #755

```
IsNumber("75") returns TRUE
IsNumber("80") returns FALSE
```

Parent topic: [Locate actions](#)

Related reference:

[IsAlpha](#)

[IsCurrency](#)

[IsDateValue](#)

[WordFind](#)

[ValueInField](#)

[RegExFind](#)

IsSelectedBlockType

Compares the provided block type to the type of the currently selected block. This action uses the document block layout that is created by the DocumentAnalytics actions.

Important: This action is currently released as a preview. If you use this action in your application, you might need to make updates to your application if a new version of this action released.

Syntax

```
bool IsSelectedBlockType (string StrParam)
```

Parameters

string StrParam

Type

The name of the expected block type. Smart parameters are supported.

Returns

True if the provided parameter matches the name of the currently selected block. Otherwise, False.

Level

Page or field.

Details

This action uses the document block layout that is created by the DocumentAnalytics actions. IsSelectedBlockType compares the provided block type to the type of the currently selected block.

Example

```
RegExFind_InBlock("Da.e")
SelectParentBlock()
IsSelectedBlockType("Line")
```

This example finds a word, selects the parent of the word and then confirms that the parent is of type `Line`.

Parent topic: [Locate actions](#)

Related reference:

[GetSelectedBlockType](#)

IsValue

Determines whether the value of the located word matches the value that is specified.

Syntax

```
bool IsValue (strParam)
```

Parameters

The value to be compared to the object's recognized value.

Returns

True if the located value matches the parameter's value. Otherwise, False.

Level

Page or field level.

Details

Using the current location of a previously located word or phrase, this action determines if the value matches the value of the input parameter. By testing the value recognized in the current word or phrase, it is possible for an application to determine it has located the data that is required, and then take subsequent actions based on the result of the test. If you want to check the value of a subset of the word or phrase, use the `ValueInWord` action. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Attention: Match test is not case sensitive and does not include leading and trailing spaces.

Example

```
WordFind("Houston")  
GoRightWord("2")  
IsValue("77770")
```

This sequence confirms that the current page's recognized value for Houston's ZIP code is "77770".

The action returns a Boolean value: True if the values are the same, False if they are not.

Parent topic: [Locate actions](#)

Related reference:

[ValueInField](#)

[WordFind](#)

[FindKeyList](#)

[RegExFind](#)

IsValue_RegEx

Determines whether the value of the located word matches the regular expression specified.

Syntax

```
bool IsValue_RegEx (strParam)
```

Parameters

A Regular Expression that will be used for comparison with the recognized value of the word or phrase.

Returns

True if the located value matches the parameter's value. Otherwise, False.

Level

Page or field level.

Details

Using the current location of a previously located word or phrase, this action determines if the regular expression provided in the input parameter finds a match in the value. By testing the value recognized in the current word or phrase, it is possible for an application to determine it has located the data that is required, and then take subsequent actions based on the result of the test. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
WordFind("Houston")  
GoRightWord("2")  
IsValue_RegEx("Total")
```

Parent topic: [Locate actions](#)

LocatePositionRestore

LocatePositionRestore returns the internal locate word pointer to the word position it was on when LocatePositionSave was called.

Syntax

```
bool LocatePositionRestore ()
```

Parameters

None

Returns

False if called on the wrong level or if the COO does not exist. Otherwise, True.

Level

Page or field level.

Details

LocatePositionRestore returns the internal locate word pointer to the word position it was on when LocatePositionSave was called.

Using LocatePositionSave and LocatePositionRestore together provides an efficient way to reset the position to a located word, when the subsequent locate actions have moved the pointer to a located word. For example, when rules need to look for text in multiple directions based on a found word, these actions efficiently reset the internal locate pointer to a saved position instead of performing the original locate action a second time, which can require more processing time.

Example

This example shows two functions. The first performs a locate, saves the location, moves to the right and tests the word. If the word does not match the IsNumber criteria, then control falls to the next function which restores the saved locate position to the word found in the previous FindKeyList action and then looks in a different direction for a numeric value. Alternatively, the second function can call FindKeyList again, but the action to restore the last position is much more efficient.

```
Key List Right 1 Function
+ FindKeyList ("InvNum")
+ LocatePositionSave ()
+ GoRightWord ("1")
+ IsNumber ("60")
+ UpdateField ()
Key List Down 1 Function
+ LocatePositionRestore ()
+ GoBelowWord ("1")
+ IsNumber ("60")
+ UpdateField ()
```

Parent topic: [Locate actions](#)

Related reference:
[LocatePositionSave](#)

LocatePositionSave

LocatePositionSave creates a restore point for the current word position of the locate pointer.

Syntax

```
bool LocatePositionSave ()
```

Parameters

None

Returns

False if called on the wrong level or if the COO does not exist. Otherwise, True.

Level

Page or field level.

Details

LocatePositionSave creates a restore point for the current word position of the locate pointer. After calling additional locate actions that change the word location of locate pointer, the previously saved position can be restored using the LocatePositionRestore action.

Using LocatePositionSave and LocatePositionRestore together provides an efficient way to reset the position to a located word, when the subsequent locate actions have moved the pointer to a located word. For example, when rules need to look for text in multiple directions based on a found word, these actions efficiently reset the internal locate pointer to a saved position instead of performing the original locate action a second time, which can require more processing time.

Example

This example shows two functions. The first performs a locate, saves the location, moves to the right and tests the word. If the word does not match the IsNumber criteria, then control falls to the next function which restores the saved locate position to the word found in the previous FindKeyList action and then looks in a different direction for a numeric value. Alternatively, the second function could call FindKeyList again, but the action to restore the last position is much more efficient.

```
Key List Right 1 Function
+ FindKeyList ("InvNum")
+ LocatePositionSave ()
+ GoRightWord ("1")
+ IsNumber ("60")
+ UpdateField ()
Key List Down 1 Function
+ LocatePositionRestore ()
+ GoBelowWord ("1")
+ IsNumber ("60")
+ UpdateField ()
```

Parent topic: [Locate actions](#)

Related reference:

[LocatePositionRestore](#)

MaxLength

Determines whether the number of characters in the located word is greater than or equal to the number specified.

Syntax

```
bool MaxLength (StrParam)
```

Parameters

An integer specifying the maximum number of characters the word or phrase can contain.

Returns

False if the parameter is not Numeric, or if the actual number of characters exceeds the parameter. Otherwise, True.

Level

Page or field level.

Details

Compares the number of characters in the located word or phrase to a maximum number you supply as the parameter. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
If the recognized value of the located word or phrase is: ANYTHING
MaxLength("14") returns TRUE
MaxLength("8") returns TRUE
MaxLength("3") returns FALSE
```

Parent topic: [Locate actions](#)

Related reference:

[MinLength](#)

MergeWordLF

Merges the located word with one or more words to the left, on the same line.

Syntax

```
bool MergeWordLF (StrParam)
```

Parameters

An integer that indicates the number of words or phrases to the left of the previously found field that is to be placed into the current object field.

Returns

Always True.

Level

Page or field level.

Details

Merges the located word or phrase with one or more words to the left, on the same line. A "word" in this context is a string of characters that might include spaces. This action is used when the value searched for might have spaces in it. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
WordFind("2000")
MergeWordLF("1")UpdateField()
```

Given the following cco string:
Invoice Date: Jan 2000

```
FindWord("2000") locates the highlighted value:
Invoice Date: Jan 2000
```

```
MergeWordLF("1") consolidates it with the text "Jan":
Invoice Date: Jan 2000
```

so that the UpdateField action will save the entire value,
"Jan 2000" into the current object's field.

Parent topic: [Locate actions](#)

Related reference:

[MergeWordRT](#)

MergeWordRT

Merges the located word with one or more words to the right, on the same line.

Syntax

```
bool MergeWordRT (StrParam)
```

Parameters

An integer indicating the number of words to the right, starting from the previously found word or phrase, to be placed into a field.

Returns

Always True.

Level

Page or field level.

Details

Places the located word or phrase with one or more words to the right, on the same line, into the current object field. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
WordFind("Jan")
MergeWordRT("1")
UpdateField()
```

For the cco string, Invoice Date: Jan 2000 , the result of `WordFind("Jan")` is to locate the highlighted word, Jan, in Invoice Date: Jan. `MergeWordRt("1")` consolidates that value with the year value to its right, which is Invoice Date: Jan 2000. `UpdateField` saves the entire value, Jan 2000, which is saved to the calling object field.

Parent topic: [Locate actions](#)

Related reference:

[MergeWordLF](#)

MinLength

Determines whether the number of characters in the located word is less than or equal to the number specified.

Syntax

```
bool MinLength (StrParam)
```

Parameters

An integer specifying the minimum number of characters the word or phrase can contain.

Returns

False if the parameter is not Numeric, or if the actual number of characters is less than the parameter. Otherwise, True.

Level

Page or field level.

Details

Compares the number of characters in the located word or phrase to a minimum number you supply as the parameter. Regardless of being called at the page or field level, this action operates on the recognized text for the current page.

Example

```
If the recognized value of the word or phrase is: ABC1  
MinLength("4") returns TRUE  
MinLength("3") returns TRUE  
MinLength("6") returns FALSE
```

Parent topic: [Locate actions](#)

Related reference:

[MaxLength](#)

RegExFind

Same as the WordFind action, except that it supports regular expressions.

Syntax

```
bool RegExFind (StrParam)
```

Parameters

A word or phrase to find on the current page. The parameter is expected to be a Regular Expression.

Returns

True if the word or phrase is located on the page. Otherwise, False.

Level

Page level.

Details

Locates the first occurrence of a word or phrase on the current page where the input search term is specified as a regular expression. The search is started from the first word on the current page. The location of the found word or phrase will be remembered so the result can be used by subsequent actions. The search is case sensitive.

Example

```
RegExFind("Da?e")
GoRightWord("1")
IsDateValue()
UpdateField()
```

RegExFind will look for the first occurrence of a word or phrase that matches the regular expression. If a match is found, it continues processing by moving one word to the right of the result of RegExFind and acts on that word. If it is a date format, UpdateField will place the value into the current field.

Parent topic: [Locate actions](#)

Related reference:

[FindRegExList](#)

[RegExFind_InZone](#)

[RegExFindNext](#)

RegExFind_InBlock

RegExFind_InBlock will look for the first occurrence of a word or phrase that matches the regular expression. This action uses the document block layout that is created by the DocumentAnalytics actions.

Important: This action is currently released as a preview. If you use this action in your application, you might need to make updates to your application if a new version of this action released.

Syntax

```
bool RegExFind_InBlock (StrParam)
```

Parameters

A word or phrase to find on the current page. The parameter is expected to be a Regular Expression. The search is case sensitive.

Returns

True if the word or phrase is located within the currently selected block. Otherwise, False.

Level

Page or field level.

Details

The action locates the first occurrence of a word or phrase on the current page where the input search term is specified as a regular expression, respecting the currently selected block of the document layout. The search is started from the first word within the currently selected block.

When the document layout is created, the entire page is set as the currently selected block, making the initial search area the entire page. Once the regular expression has been matched, the location of the found word or phrase will be remembered so the result can be used by subsequent actions.

Additionally, the found word will become the currently selected word and the block region of that word becomes the currently selected block area within the document block layout. Subsequent block actions will act upon the currently selected block, allowing the area of interest to be changed to control the area of the page that is the target of subsequent actions.

Example

```
RegExpFind_InBlock("Da.e")
SelectParentBlock()
GoRightWord("1")
IsDateValue()
UpdateField()
```

RegExpFind_InBlock will look for the first occurrence of a word or phrase that matches the regular expression. If a match is found, the next action will change the currently selected block from the found word to the entire line that contains the word, then move to the right of the word. If it is a date format, UpdateField will place the value into the current field.

Parent topic: [Locate actions](#)

RegExpFind_InZone

Same as the RegExpFind action, except that it searches the current field only.

Syntax

```
bool RegExpFind_InZone (StrParam)
```

Parameters

A word or phrase to find in the current field. The parameter is expected to be a Regular Expression.

Returns

True if the word or phrase is located in the current field. Otherwise, False.

Level

Field level.

Details

Locates the first occurrence of a word or phrase in the current field where the input search term is specified as a regular expression. The search is started from the first word of the current field. The location of the found word or phrase will be remembered so the result can be used by subsequent actions. The search is case sensitive.

Example

```
RegExpFind_InZone ("Da?e") GoRightWord ("1")
IsDateValue ()
UpdateField ()
```

RegExpFind_InZone will look for the first occurrence of a word or phrase that matches the regular expression. If a match is found, it will continue processing by moving one word right of the result of RegExpFind and act on that word. If it is a date format, UpdateField places the value into the current field.

Parent topic: [Locate actions](#)

Related reference:

[FindRegExpList](#)

[RegExpFindNext_InZone](#)

RegExpFindNext

Same as the WordFindNext action, except that it supports regular expressions.

Syntax

```
bool RegExpFindNext (StrParam)
```

Parameters

A word or phrase to find on the current page. The parameter is expected to be a Regular Expression.

Returns

True if the word or phrase is located on the page. Otherwise, False.

Level

Page level.

Details

Starting from the location of a previously found word or phrase, this action locates the first occurrence of a word or phrase on the current page where the input search term is specified as a regular expression. The search is started from the location of a previously found word or phrase. The location of the found word or phrase will now be remembered so the result can be used by subsequent actions. The search is case sensitive.

Example

```
RegExpFind ("ItemID")
RegExpFindNext ("Desc")
```

In this sequence, the first action looks for ItemID. If the search succeeded, RegExFindNext looks for the first occurrence of Desc that comes after ItemID.

Parent topic: [Locate actions](#)

Related reference:

[FindRegExList](#)

[RegExFind_InZone](#)

[RegExFindNext](#)

RegExFindNext_InBlock

Starting from the location of a previously found word or phrase, this action locates the first occurrence of a word or phrase in the currently selected block where the input search term is specified as a regular expression. Once the regular expression has been matched, the location of the found word or phrase will be remembered so the result can be used by subsequent actions. This action uses the document block layout that is created by the DocumentAnalytics actions.

Important: This action is currently released as a preview. If you use this action in your application, you might need to make updates to your application if a new version of this action released.

Syntax

```
bool RegExFindNext_InBlock (StrParam)
```

Parameters

A word or phrase to find in the current field. The parameter is expected to be a Regular Expression. The search is case sensitive.

Returns

True if the word or phrase is located within the currently selected block. Otherwise, False.

Level

Page or field level.

Details

Starting from the location of a previously found word or phrase, this action locates the first occurrence of a word or phrase in the currently selected block where the input search term is specified as a regular expression. Once the regular expression has been matched, the location of the found word or phrase will be remembered so the result can be used by subsequent actions. Additionally, the found word becomes the currently selected word and the block region of that word becomes the currently selected block area within the document block layout. Subsequent block actions act upon the currently selected block, allowing the area of interest to be changed to control the area of the page that is the target of subsequent actions.

Example

```
RegExFind_InZone("ItemID")
SelectParentBlock()
GoParentBlockNext()
RegExFindNext_InZone("Description")
```

```
GoRightWord("1")
UpdateField()
```

In this sequence, the first action looks for `ItemID` starting from the beginning of the currently selected block. If the search succeeded, the selected block is changed to the containing line, then changed to the following line. `RegExFindNext_InZone` looks for the first occurrence of `Description` within the line that follows `ItemID`, selects the word that follows `ItemID`, and updates the current field.

Parent topic: [Locate actions](#)

Related reference:
[RegExFind_InBlock](#)

RegExFindNext_InZone

Same as the `RegExFindNext` action, except that it searches the current field only.

Syntax

```
bool RegExFindNext_InZone (StrParam)
```

Parameters

A word or phrase to find in the current field. The parameter is expected to be a Regular Expression.

Returns

True if the word or phrase is located on the page. Otherwise, False.

Level

Page level.

Details

Starting from the location of a previously found word or phrase, this action locates the first occurrence of a word or phrase in the current field where the input search term is specified as a regular expression. The search is started from the location of a previously found word or phrase in the current field. The location of the found word or phrase will be remembered so the result can be used by subsequent actions. The search is case sensitive.

Example

```
RegExFind_InZone ("ItemID")
RegExFindNext_InZone ("Desc")
```

In this sequence, the first action looks for `ItemID`. If the search succeeded, `RegExFindNext` looks for the first occurrence of `Desc` that comes after `ItemID`.

Parent topic: [Locate actions](#)

Related reference:
[FindRegExList](#)
[RegExFind_InZone](#)
[RegExFindNext](#)

ScanRT

Moves the specified number of words to the right of the current word. Expands the search area up and down slightly in case the word is a little above or below the current word.

Syntax

```
bool ScanRT (StrParam)
```

Parameters

Numeric value of the number of words to be evaluated to the right of the current word or phrase.

Returns

True, if a word is found. Otherwise, False.

Level

Field level only.

Details

ScanRT (scan right) looks for a word in positions that are slightly above or below the line on which the current word or phrase is located.

Example

```
WordFind("Number")
ScanRT("1")
```

In this example, the action finds the `Number` on the current page. Then, it moves one word to the right as it searches for a value.

To compensate for the possibility that this value might be printed slightly above or below the line on which `Number` was printed, the ScanRT action expands the area for the target value. When a page is recognized, then skewing or data layout might cause alignment problems. Words might not be recognized as being on the same line and might be recognized as being slightly above or below the current line.

This action compensates for alignment problems at the current word location. The action uses a calculation adjustment to determine which of the words to the right of the current word can be considered to be on the same line. It then remembers the location of the word that best fits this criteria. This new remembered location can then be used by subsequent actions.

Parent topic: [Locate actions](#)

Related reference:

[GoRightWord](#)

SelectParentBlock

Looks for the first occurrence of a word or phrase that matches the regular expression. If a match is found, the next action will change the currently selected block from the found word to the entire line that contains the

word. SelectParentBlock is then called again to change the selected block to the parent of the current line, then the current field zone is updated to contain the entire set of lines contained in the current block and the field value is set to all of the text within the current block. This action utilizes the document block layout created by the DocumentAnalytics actions.

Important: This action is currently released as a preview. If you use this action in your application, you might need to make updates to your application if a new version of this action released.

Syntax

```
bool SelectParentBlock ()
```

Parameters

None.

Returns

True if the word or phrase is located on the page. Otherwise, False.

Level

Page or field level.

Details

The currently selected block, which denotes an area on the page, is changed to the containing block. This effectively makes the new area of interest larger than the previous area.

For example, if the currently selected block is a word, when SelectParentBlock is called, the selected word remains the same, but the area of interest is changed to the parent block, which would be the entire line. Subsequent actions that act on the current block now respects the line as the area of interest on the page.

Calling SelectParentBlock again moves the selected block to the parent of the line. Determining which block gets selected depends on the layout of the page. One scenario is that the selected block will now be a paragraph that contains multiple lines, including the word that was previously found. Any block actions, such as additional searches, will be run within the boundaries of this newly selected paragraph block.

While the block nesting layout often follows the same pattern, the layout of blocks will depend on the layout of text on the page. For predictable results, the application rules might need to work on expected layouts based on the page types that are supported by the application.

Example

```
RegexFind_InBlock("Da.e")  
SelectParentBlock()  
SelectParentBlock()  
UpdateFieldWithBlock()
```

RegexFind_InBlock will look for the first occurrence of a word or phrase that matches the regular expression. If a match is found, the next action will change the currently selected block from the found word to the entire line that contains the word. SelectParentBlock is then called again to change the selected block to the parent of the current line, then the current field zone is updated to contain the entire set of lines contained in the current block and the field value is set to all of the text within the current block.

Parent topic: [Locate actions](#)

SelectParentBlockOuterType

Selects the outermost parent of the current block of the type that is specified. If the currently selected block has multiple parent blocks of the same type, then the parent farthest from the current block is selected. This action uses the document block layout that is created by the DocumentAnalytics actions.

Important: This action is currently released as a preview. If you use this action in your application, you might need to make updates to your application if a new version of this action released.

Syntax

```
bool SelectParentBlockOuterType (string selectType)
```

Parameters

string selectType

selectType

The type of the outer most parent block to select.

Smart parameters are supported.

Returns

True if the word or phrase is located on the page. Otherwise, False.

Level

Field or page level.

Details

This action is usually the last rule of a Locate RuleSet, and is used when the probable area of the sought after value has been found.

Example

```
RegExFind_InBlock("Da.e")  
SelectParentBlockOuterType("Block")
```

In this example the selected block will be the one of type `Block` that is the farthest parent to the block containing the found word. The available block types are: Document, Page, Title, Block, Table, Row, Cell, Paragraph, Header, and Footer. Consider a page that contains the following nested block layout:

```
[Page]  
[Block]  
[Block]  
[Line]  
[Word] "Date" [/Word]  
[/Line]  
[/Block]  
[/Block]  
[/Page]
```

In this example, the selected block would be the `Block` that is the first child to the `Page` block.

Parent topic: [Locate actions](#)

Related reference:

[SelectParentBlock](#)

[SelectParentBlockType](#)

SelectParentBlockType

Selects the parent of the current block of the type that is specified. If the currently selected block has multiple parent blocks of the same type, then the parent nearest to the current block would be selected. Uses the document block layout created by the DocumentAnalytics actions.

Important: This action is currently released as a preview. If you use this action in your application, you might need to make updates to your application if a new version of this action released.

Syntax

```
bool SelectParentBlockType (string selectType)
```

Parameters

string selectType

selectType

The type of the parent block to select.

Smart parameters are supported.

Returns

True the specified parent block is found and selected. If the current block is of the specified type, then true will be returned. Otherwise, False.

Level

Page or field level.

Details

If the current block is of the specified type, then the action will return true and the selected block will remain at the current block. When starting a new search after performing previous searches, it may be useful to call this action with type `Page` to reset the selected block to the page block so the entire page block area can be searched again using the block actions.

Example

```
RegexFind_InBlock("Da.e")  
SelectParentBlockType("Block")
```

In this example the selected block will be the one of type `Block` that is the nearest parent to the block containing the found word. The available block types are: Document, Page, Title, Block Table, Row, Cell, Paragraph, Header, and Footer. Consider a page that contains the following nested block layout:

```
[Page]
[Block]
[Block]
[Line]
[Word] "Date" [/Word]
[/Line]
[/Block]
[/Block]
[/Page]
```

In this example, the selected block would be the `Block` that is the parent to the `Line` block.

`SelectParentBlockType("Page")`

Specifying `Page` selects the page block as the currently selected block. When a previous `Locate` action has been run, reselecting the `Page` block will reset the block to the page so that the next action that searches within the selected block will search the entire page.

Parent topic: [Locate actions](#)

Related reference:

[SelectParentBlock](#)

[SelectParentBlockOuterType](#)

SelectSnippet

Populates a snippet field with the recognized value of the located word. Used with directional actions.

Syntax

```
bool SelectSnippet (strParam)
```

Parameters

1. The character that is to appear in the Snippet field if a value is not available. "~" is the default.
2. The size of the captured value's snippet image. "1" is 1x of the actual word area, and is the default. "2" is 2x of the actual word area, the zone is twice the size of the selected target word. ".5" is 1/2 the actual word area, not recommended as only a portion of the target word would display in the snippet at verify time.

Returns

Always True.

Level

Field level only.

Details

Used in conjunction with directional actions, this action will populate a Snippet field with the recognized value of the located word or phrase.

This action is usually the last rule of a `Locate RuleSet`, and is used when the probable area of the sought after value has been found.

Example

```
FindKeyInList ("InvNum")
GoRightWord ("1")
SelectSnippet ("~",1")
```

This sequence first tries to locate an Invoices Number keyword in the current page. If successful, the next action attempts to lock onto a word or phrase to the right of the located word or phrase.

If that word is present, the SelectSnippet action will place the image of the word's recognized value into the Snippet of the applicable Field object. The Data Entry operator can then determine if the Snippet contains the correct value and can enter the data into the accompanying Data Edit field in the Data Entry Panel.

Parent topic: [Locate actions](#)

SetKeyFileEncodingAsUnicode

Determines if the Locate key files are opened as ASCII or UNICODE.

Member of namespace

Locate

Syntax

```
bool SetKeyFileEncodingAsUnicode (StrParam)
```

Parameters

- True - Key files open as UNICODE encoding.
- False - Key files opens as ASCII encoding and is Default.

Smart parameters are supported.

Returns

Always True.

Level

Any.

Details

Determines if the Locate key files are opened as ASCII or UNICODE. This action must be called before any Locate action that uses a key file. If not called, then the key files are opened as ASCII.

Example:

```
SetKeyFileEncodingAsUnicode ("True")
```

Parent topic: [Locate actions](#)

SetRect

Sets the position and size of the current field in the page data file to the values specified.

Syntax

```
bool SetRect (strParam)
```

Parameters

Four comma separated coordinates designating the rectangle's size and location: X, Y, Width and Height.

Returns

Always True.

Level

Field level only.

Details

Updates the zone of the current field.

Example

```
SetRect ("0,0,100,200")
```

This action changes the field's zone position to a rectangle at coordinates 0, 0, which is 100 wide and 200 high.

This action is useful if you wish to change the zone to a specific area of the image where you know your value resides.

Parent topic: [Locate actions](#)

SetVirtualPageEndPosition

Use this action after you use the Locate commands that identify the last word of a new virtual page.

Syntax

```
bool SetVirtualPageEndPosition()
```

Parameters

None.

Returns

True, if the action is successful. False, if an error occurs.

Level

Page level or field level.

Details

This action requires that the page has an associated layout file and the CCO populated by using the `CreateCcoFromLayout` action. This action is expected to be used with `CreateVirtualPage` to build a new page by using a subset of text from the current page. For more information, see [CreateVirtualPage](#).

Example

```
Recognize ()
CreateCcoFromLayout ()
WordFind ("Hello")
SetVirtualPageStartPosition ()
WordFindNext ("Good bye")
GoUpLine ("1")
GoLastWord ()
SetVirtualPageEndPosition ()
CreateVirtualPage ("MyNewPageType")
```

Parent topic: [Locate actions](#)

SetVirtualPageStartPosition

Use this action after you use the Locate commands that identify the first word of a new virtual page.

Syntax

```
bool SetVirtualPageStartPosition ()
```

Parameters

None.

Returns

True, if the action is successful. False, if an error occurs.

Level

Page level or field level.

Details

This action requires that the page has an associated layout file and the CCO populated by using the `CreateCcoFromLayout` action. This action is expected to be used with `CreateVirtualPage` to build a new page by using a subset of text from the current page. For more information, see [CreateVirtualPage](#).

Example

```
Recognize ()
CreateCcoFromLayout ()
WordFind ("Hello")
SetVirtualPageStartPosition ()
WordFindNext ("Good bye")
GoUpLine ("1")
GoLastWord ()
```

```
SetVirtualPageEndPosition()  
CreateVirtualPage("MyNewPageType")
```

Parent topic: [Locate actions](#)

UpdateDCOField

Updates the position coordinates for the specified field in the page data file with the position of the located word.

Syntax

```
bool UpdateDCOField (StrParam)
```

Parameters

String value of the target field's name (Smart Parameter enabled), as a Field object of the Document Hierarchy.

Returns

False if the source page's Fingerprint file (.cco) is not available or is empty; if the target field cannot be found; or if there is no information about the target field's starting and ending sizes and locations. Otherwise, True.

Level

Page or Field level.

Details

This action updates the size and position coordinates of the Field object representing the field identified by the parameter. Typically, the action follows earlier actions and events which modify the field's width and height, or its precise placement on the current source page.

Do not confuse this action with the UpdateField action, which updates Text values. This action does not update a field's Text value. Instead, it modifies the size and location parameters of a field or zone.

Example

```
To find a sibling field 'Preparer_Name':  
UpdateDCOField("../Preparer_Name")
```

```
To find a child field 'Preparer_Name':  
UpdateDCOField("Preparer_Name")
```

Parent topic: [Locate actions](#)

UpdateField

Updates the current field in the page data file with the value and position of the located word.

Syntax

```
bool UpdateField ()
```

Parameters

None.

Returns

Always True.

Level

Field level only.

Details

Updates the appropriate field in the current page's Data file with the recognized (and possibly formatted) value of the located word or phrase.

Important: An entered value that the UpdateField action places in a Data file becomes a captured value, and can be processed by Validation and Export RuleSets.

Example

```
FindKeyList("Date")
GoRightWord("1")
IsDateValue()
UpdateField()
```

The first action in the sequence finds a word or phrase that identifies the Date Field object of the current page. This is the field's static value - probably its title.

The next action moves right one word or phrase to locate the field's entered value, a recognized date such as 12/31/2002. The third action checks to be sure the value has an acceptable Date format.

The concluding UpdateField action takes place only if the others are successful. It adds the field's entered value to the current page's Data file, where it is a captured value awaiting the attention of upcoming rules with Validate and Export actions.

Parent topic: [Locate actions](#)

UpdateFieldWithBlock

Updates the current DCO object's field zone coordinates with the coordinates of the currently selected block. The text of the field will be set with the contents of the text bound by the currently selected block in the document layout. Uses the document block layout created by the DocumentAnalytics actions.

Important: This action is currently released as a preview. If you use this action in your application, you might need to make updates to your application if a new version of this action released.

Syntax

```
bool UpdateFieldWithBlock ()
```

Parameters

None.

Returns

True if the word or phrase is located on the page. Otherwise, False.

Level

Field level only.

Details

UpdateFieldWithBlock updates the current DCO object's field zone coordinates with the coordinates of the currently selected block. The text of the field will be set with the entire text within the currently selected block in the document layout.

Example

```
RegExFind_InBlock("Da.e")
SelectParentBlock()
SelectParentBlock()
UpdateFieldWithBlock()
```

RegExFind_InBlock looks for the first occurrence of a word or phrase that matches the regular expression. If a match is found, the next action will change the currently selected block from the found word to the entire line that contains the word. SelectParentBlock is then called again to change the selected block to the parent of the current line, likely a paragraph. Then the current field zone is updated to contain the entire set of lines contained in the current block and the field value is set to all of the text within the current block.

Parent topic: [Locate actions](#)

Related reference:

[SelectParentBlock](#)

[SelectParentBlockType](#)

ValueInField

Determines whether any part of the located word matches the value that is specified.

Syntax

```
bool ValueInField (StrParam)
```

Parameters

The value that is to be matched with a portion of the value in the current field.

Returns

False if no match occurs. Otherwise, True.

Level

Field level.

Details

Checks if the parameter you enter is within the value of the current field represented by the bound Field object of the Document Hierarchy. Only a portion of the field's value needs to match the parameter. If the entire field must match, use `IsValue`. Case insensitive.

Example

```
ValueInField("Invoice")
```

Parent topic: [Locate actions](#)

Related reference:

[IsValue](#)

[ValueInField_RegEx](#)

ValueInField_Fuzzy

Uses fuzzy matching to determine whether any part of the located word matches the value that is specified.

Syntax

```
bool ValueInField_Fuzzy (StrParam)
```

Parameters

String value to be matched to the current field's value, using fuzzy matching procedures.

Returns

False if no match occurs. Otherwise, True.

Level

Field level only.

Details

Checks if there is a "fuzzy" match of the parameter's value with the value in the current field. Only a portion of the field's value needs to match. The match is performed by allowing for common substitutions that can occur during recognition. These substitutions include characters: B8, Z2, S5, oO0 and iItl1. Case insensitive.

Example

```
ValueInField_Fuzzy("Invoice")
```

Parent topic: [Locate actions](#)

ValueInField_RegEx

Determines whether any character or series of characters in the located word matches the specified regular expression.

Syntax

```
bool ValueInField_RegEx (StrParam)
```

Parameters

The portion of the value to find in the field. The parameter is expected to be expressed as a regular expression.

Returns

False if no match occurs. Otherwise True.

Level

Field level only.

Details

This action checks if the Regular Expression you specify as the parameter is equivalent to the value of the current field. Only a part of the field must match the parameter. To match the entire value of the field, use `IsValueRegEx`.

Example

```
ValueInField_RegEx("[\^\b\s\n\r]Inv[oO][iItl1]ce[\b\s]*")
```

Parent topic: [Locate actions](#)

Related reference:

[IsValue](#)

[IsValue_RegEx](#)

[ValueInField](#)

WordFind

Locates the first or next occurrence of the specified word or phrase on the current page.

Syntax

```
bool WordFind (StrParam)
```

Parameters

A word or phrase to find on the page.

Returns

True, if the word or phrase is on the page. Otherwise, False.

Level

Field level only.

Details

The current page is searched to find the whole word or phrase. The location of the first word or phrase that matches the parameter is remembered and can be used by subsequent actions. Matching is case-sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes `will` and the recognition that is read is `wi11`, a match still occurs.

This action matches whole word values and does not search for parts of words or characters within words.

Common substitutions include characters: B8, Z2, S5, o00, and i1t1

Example

```
WordFind("Sales Tax")
GoRightWord("1")
IsCurrency()
UpdateField
```

In this example, the `WordFind` action looks for the first occurrence of `Sales Tax` within the current page, always starting at the first word of the page. If the phrase `Sales Tax` is found, the subsequent actions operate based on the location of the found phrase.

Parent topic: [Locate actions](#)

Related reference:

[WordFindNext](#)

[FindKeyList](#)

[AddKeyList](#)

WordFind_InZone

Same as the `WordFind` action, except that it searches the current field only.

Syntax

```
bool WordFind_InZone (StrParam)
```

Parameters

A word or phrase to find in the current zoned field.

Returns

True if the word or phrase is located in the field. Otherwise, False.

Level

Page or Field

Details

The current field will be searched to find the word or phrase. The location of the first word or phrase that matches the parameter will be remembered so it can be utilized by subsequent actions. Matching is case sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes "will" and the recognition read "wi11", a match will still occur.

Common substitutions include characters: B8, Z2, S5, oO0 and iIt1.

Example

```
WordFind_InZone("Sub Total")
GoRightWord("1")
IsCurrency()
```

In this example, the WordFind action looks for the first occurrence of Sub Total within the current field, always starting at the first word of the zone. If the phrase Sub Total is found, the subsequent actions will operate based on the location of the found phrase.

Parent topic: [Locate actions](#)

Related reference:

[WordFind](#)

[WordFindNext](#)

[WordFindNext_InZone](#)

WordFindNext

Same as the WordFind action, except that it locates the next occurrence.

Syntax

```
bool WordFindNext (StrParam)
```

Parameters

A word or phrase to find on the page, following a previously found word or phrase.

Returns

True if the parameter is located. Otherwise, False.

Level

Field level only.

Details

The current page will be searched to find the word or phrase, starting from the location remembered from the last search, such as from WordFind. The location of the first word or phrase that matches the parameter will now be remembered so it can be utilized by subsequent actions. Matching is case sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes "will" and the recognition read "wi11", a match will still occur.

Common substitutions include characters: B8, Z2, S5, oO0 and iIt1.

Example

```
WordFindNext ("Total")
WordFindNext ("Total")
GoRightWord ("1")
IsCurrency ()
UpdateField
```

In this sequence, the Locate rule locates the third instance of the word Total in the current page.

Parent topic: [Locate actions](#)

Related reference:

[WordFind](#)

[FindLastWord](#)

WordFindNext_InZone

Same as the WordFindNext action, except that it searches the current field only.

Syntax

```
bool WordFindNext_InZone (StrParam)
```

Parameters

A word or phrase to find in the current field, following a previously found word or phrase.

Returns

True if the parameter is located. Otherwise, False.

Level

Field level.

Details

The current field will be searched to find the word or phrase, starting from the remembered location of a previous find. The new location of the first word or phrase in the field that matches the parameter will now be remembered so it can be utilized by subsequent actions. Matching is case sensitive.

To improve matching, this action automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes "will" and the recognition read "wi11", a match will still occur.

Common substitutions include characters: B8, Z2, S5, o00 and i1t1.

Example

```
WordFind_InZone (ItemID)
WordFindNext_InZone (Desc)
```

In this example, the WordFind_InZone action looks for the first occurrence of ItemID within the current field, always starting at the first word of the zone. If the phrase ItemID is found, the word Desc will be looked for in the current field, starting after the location of ItemID. If Desc is found, any subsequent actions will operate based on the location of that found phrase.

Parent topic: [Locate actions](#)

Related reference:

[WordFind_InZone](#)
[FindLastWord_InZone](#)

WordFind_Offset

Sets the value of the page's Image_Offset variable. The variable is based on the difference in the position of the specified word on the current page and on the matched fingerprint image.

Syntax

```
bool WordFind_Offset (StrParam)
```

Parameters

1. String value of a keyword that the action is to find on both the fingerprint and recognized image.
2. An optional parameter that specifies the offset threshold. If not specified, the default value is 100 pixels.

Returns

Always True.

Level

Page level.

Details

This action locates a word or phrase on both the recognized page and on the fingerprint. The positioning of both locations are compared to determine an offset value. The calculated difference is stored in the DCO of the current page in the variable Image_Offset. This value will be used by subsequent actions, such as ReadZones, to compensate for the difference so the field data is properly located.

For best results, the word or phrase should appear only once or the first instance of the word or phrase should always appear in the same location.

The threshold is the maximum distance between keywords found on the Live image and the fingerprint. The Default value is 100 pixels. If the keywords are more than 100 pixels apart no Offset is generated; preventing matches of keywords where there is more than one instance of the word to be found on an image. A successful matched pair will update the Image_Offset variable at the page level.

This action requires the FingerPrint CCO to have full page recognition results. Otherwise, there will be nothing to match against.

This action should not be located in the same ruleset where full page recognition is done. It needs to exist in a ruleset that is performed afterwards.

Example

```
WordFind_Offset ("Invoice")
```

Parent topic: [Locate actions](#)

Lookup actions

Use the Lookup actions to validate field values by using database lookups and populate fields with lookup results.

The Lookup actions opens a connection to the Lookup database and populates the current field with a value returned by the previous ExecuteSQL or LookupReturnValue action.

- [ClearLookupResults](#)
Clears the results that are returned by the previous Lookup action.
- [CloseConnection](#)
Closes an open connection to the Lookup database.
- [ExecuteSQL](#)
Runs a SQL statement on the Lookup database. If a SELECT statement returns one or more values, these values are stored in an internal data record that you can access by using the PopulateWithResult action.
- [ExecuteSQLEx](#)
Executes the SQL statement you enter in the first parameter.
- [OpenConnection](#)
Uses a data source name or connection string to open a connection to a Lookup database.
- [PopulateWithResult](#)
Populates the current field with a value returned by the previous ExecuteSQL or LookupReturnValue action. If the previous action returned multiple values, you can specify the value that you want to use.
- [SmartSQL](#)
Runs a SQL statement that supports smart parameters.
- [SmartSQLEx](#)
Executes the SQL statement you enter in the first parameter.

Parent topic: [Global actions](#)

ClearLookupResults

Clears the results that are returned by the previous Lookup action.

Syntax

```
bool ClearLookupResults ()
```

Parameters

None.

Returns

Always True.

Level

All.

Details

This action clears the stored results returned from a previous Lookup action such as PopulateWithResult.

Example:

```
OpenConnection("@APPVAR(* /lookupdb:cs)")
ExecuteSQL("SELECT NAME, ADDRESS FROM Vendor;")
PopulateWithResult("1")
ClearLookupResults()
```

Here, ClearLookupResults() clears the stored results of the Vendor Name and Address.

Parent topic: [Lookup actions](#)

CloseConnection

Closes an open connection to the Lookup database.

Syntax

```
bool CloseConnection ()
```

Parameters

None.

Returns

True, even if the connection is already closed.

Level

All, but generally used as part of a separate RuleSet at the Batch level.

Details

Closes an open connection to your Lookup database.

Usually, this action is placed in a RuleSet that is separate from the RuleSet that opens the connection and stores the data.

LookupDBCclose RuleSet, for example, is run at the Batch level after all data has been exported from the batch to the specified database.

Example:

```
CloseConnection()
```

This action closes the previously opened connection to the Lookup database. This action is usually part of a separate RuleSet that prevents the need to repeatedly open the connection to the database. You can open the connection once in the first RuleSet, use the database from all documents and pages in the batch, then close the connection once in the second RuleSet.

Parent topic: [Lookup actions](#)

Related reference:

[OpenConnection](#)

ExecuteSQL

Runs a SQL statement on the Lookup database. If a SELECT statement returns one or more values, these values are stored in an internal data record that you can access by using the `PopulateWithResult` action.

Syntax

```
bool ExecuteSQL (sStringIn)
```

Parameters

1. The SQL expression that you want to run surrounded by quotation marks (" "). '%s' or %s can be substituted in the SQL expression to represent a field value.
 - o If %s represents a text field, it must be surrounded by ' '.
 - o If %s represents a numeric field, it appears without surrounding apostrophe characters.
2. , 3+ Field names whose captured values you want to use in the SQL expression (see the example).

Returns

True, if the SQL statement runs successfully, select statements must also return a value. False, if it does not.

Level

All.

Details

Runs the SQL statement that you enter in the first parameter. Field values can be substituted for "%s" in the statement.

Example:

This sequence opens a connection to the InvoiceLook database. Next, it inserts values into the `CompanyCode` and `Type` columns of the `Vendor` table:

```
OpenConnection("@APPVAR(* /lookupdb:cs)")
ExecuteSQL("INSERT INTO Vendor (CompanyCode,Type) VALUES ('MQSW','New')")
```

Here, `dbVendorID` is a numeric field, while `dbVendorname` is a text field.

```
OpenConnection("@APPVAR(* /fingerprintconn:cs)")
ExecuteSQL("""SELECT CompanyCode FROM Vendor WHERE dbVendorID = %s AND
dbVendorName = '%s';",VendorID,VendorName")
```

Parent topic: [Lookup actions](#)

ExecuteSQLEx

Executes the SQL statement you enter in the first parameter.

Member of namespace

Lookup

Syntax

```
bool ExecuteSQLEx (string sStringIn, bool sPopulate, string sTarget)
```

Parameters

- **sStringIn** - The SQL expression you want to execute surrounded by quotation marks (" ")
%s' or %s can be substituted in the SQL expression to represent a field value.
If %s represents a text field, it must be surrounded by ' '.
If %s represents a numeric field, it appears without surrounding apostrophe characters **2,3+**.
Field names whose captured values you want to use in the SQL expression (see the example).
- **sPopulate** - True/False value to enable or suppress populating value of any record returned from the SQL expression.
- **sTarget** - Smart Parameter destination for value population, leave blank for current object's Text property.

Returns

True if the SQL statement executes successfully, selected statements must also return a value.

False, if it does not.

Level

All.

Details

Executes the SQL statement you enter in the first parameter. Field values can be substituted for "%s" in the statement.

Example:

```
ExecuteSQLEx("INSERT INTO Vendor (CompanyCode,Type) VALUES ('MQSW','New')",  
"False", "")
```

Parent topic: [Lookup actions](#)

OpenConnection

Uses a data source name or connection string to open a connection to a Lookup database.

Syntax

```
bool OpenConnection (strParam)
```

Parameters

The Database Connection String, using an OLEDB or ODBC database provider.

If the action is establishing a connection with DB2, Oracle database, or SQL Server database, you must specify the user ID and Password. Some database providers support integrated authentication, which use the credentials of the running process.

Smart parameters are supported and recommended to prevent plain text credentials in the application rules.

Returns

True if the action results in a connection to the database. Otherwise, False.

Level

All.

Details

This action uses the Connection String that you provide as the parameter to open a connection to your LookUp database.

Example:

```
OpenConnection("@APPVAR(* /fingerprintconn:cs ")
```

This example opens the fingerprint database and obtains the connection information from the Application Service. It is recommended to use the application service so passwords are kept hidden.

Access:

```
OpenConnection("Provider=MSACCESS;DSN=C:\Datacap\1040EZ\1040ezLook.mdb;UID=;PWD=;")
```

Oracle:

```
OpenConnection("Provider=OraOLEDB.Oracle.1;Password=myspassword;Persist Security Info=True;User ID=myuserid;Data Source=TM2")
```

SQL Server (Standard authentication):

```
OpenConnection("Provider=SQLOLEDB;Data Source=SQLServerName;Initial Catalog=myDatabase;User ID=myuserid;Password=myPassword;")
```

SQL Server (Integrated authentication):

```
OpenConnection("Provider=SQLOLEDB;Data Source=SQLServerName;Initial Catalog=myDatabase;Integrated Security=SSPI;")
```

DB2:

```
OpenConnection("Provider=IBMDADB2;Hostname=1.22.333.444;Port=50000;Data Source=DB2;Database=myDatabase;User ID=myuserid;Password=myPassword;")
```

Parent topic: [Lookup actions](#)

Related reference:

[CloseConnection](#)

PopulateWithResult

Populates the current field with a value returned by the previous ExecuteSQL or LookupReturnValue action. If the previous action returned multiple values, you can specify the value that you want to use.

Syntax

```
bool PopulateWithResult (StrParam)
```

Parameters

1. A number that indicates the value in a record that is retrieved by an earlier ExecuteSQL or SmartSQL action is to be assigned to the current Field object (and added to the Data file of the current page.)
 - o "1" refers to the first column in a recordset, "2" refers to the second column, and so on.
2. True or False. True causes the action to fail, if the action returns a recordset with multiple lines. False permits the action to accept a record set with multiple lines but to use values in the first record of the record set.

Returns

True, if the second parameter is True and a previous ExecuteSQL or SmartSQL action finds a recordset with only one record.

True, if the second parameter is "False" and a previous ExecuteSQL (or SmartSQL) action finds a recordset with one or more records.

Otherwise, False.

Level

Field level only.

Details

Populates a Field object with a database value retrieved by a ExecuteSQL or SmartSQL action.

This action allows multiple rules to populate multiple Field objects with data from a single database record (see the following example.)

Example:

```
(Field #1)
OpenConnection("@APPVAR(* /lookupdb:cs) ")
ExecuteSQL("Select * From Vendor Where VendorID = %s;", VendorID")
PopulateWithResult("1, FALSE")
```

```
(Field #2)
PopulateWithResult("2, FALSE")
```

In the example, the ExecuteSQL action of the RuleSet applied to Field #1 retrieves the recordset (if it exists).

The PopulateWithResult action places the value of the first record's first column into the field where the rule applied. The PopulateWithResult action of a rule that is applied to Field #2 populates the field with the value of the first record's second column.

False means that the action can accept a recordset with multiple records but extracts values from the first record only.

Parent topic: [Lookup actions](#)

Related reference:

[ExecuteSQL](#)

[SmartSQL](#)

SmartSQL

Runs a SQL statement that supports smart parameters.

Syntax

```
bool SmartSQL (string sStringIn, string sPopulate)
```

Parameters

1. The SQL expression that you want to run. Smart parameters are supported within the expression.
2. True/False value to enable or suppress populating value of any record that is returned from the SQL expression.

Returns

True, if the SQL statement runs successfully, select statements must also return a value. False, if the SQL statement does not execute successfully.

Level

All.

Details

Runs the SQL statement that you enter in the first parameter.

Example:

This sequence opens a connection to the InvoiceLook database. Next, it inserts values into the CompanyCode and Type columns of the Vendor table:

```
OpenConnection("@APPVAR(* /lookupdb:cs ")
SmartSQL("INSERT INTO Vendor (CompanyCode,Type) VALUES ('MQSW','New')")
```

Here, dbVendorID is a numeric field, while dbVendorname is the calling text field.

```
OpenConnection("@APPVAR(* /lookupdb:cs ")
SmartSQL("SELECT CompanyCode FROM Vendor WHERE dbVendorID =+@P\VendorID+ AND
dbVendorName = '+@F+';", YES)
```

Parent topic: [Lookup actions](#)

SmartSQLEx

Executes the SQL statement you enter in the first parameter.

Member of namespace

Lookup

Syntax

```
bool SmartSQLEx (string sStringIn, bool sPopulate, string sTarget)
```

Parameters

- **sStringIn** - The SQL expression you want to execute. Smart parameters are supported within the expression.
- **sPopulate** - True/False value to enable or suppress populating value of any record returned from the SQL expression.
- **sTarget** - Smart Parameter destination for value population, leave blank for current object's Text property.

Returns

True if the SQL statement executes successfully, selected statements must also return a value.

False, if the SQL statement does not execute successfully.

Level

All.

Details

Executes the SQL statement you enter in the first parameter.

Example:

```
SmartSQLEx("INSERT INTO Vendor (CompanyCode,Type) VALUES ('MQSW','New')",  
"False", "")
```

Parent topic: [Lookup actions](#)

MC_Identify

Use the MC_Identify actions to identify claim forms in a batch.

The MC_Identify actions identify the medical claim forms that are processing in the batch.

- [AutoField](#)
Identifies red HCFA-1500 or red UB04 forms.
- [FindFields](#)
Sets up a data file for the current page and provides field position information to the data file.
- [ReadDCOSetup](#)
Designates the file name of the document hierarchy.
- [ReadPageSetup](#)
Designates the file name and path of the document hierarchy.
- [SetFormType](#)
Set the value of the Form Type that is used by the AutoField action.
- [SetMaxTolerantDistance](#)
Set the tolerance level that the AutoField action uses to match HCFA-1500 or red UB04 forms.

Parent topic: [Global actions](#)

Parent topic: [Medical Claims actions](#)

AutoField

Identifies red HCFA-1500 or red UB04 forms.

Member of namespace:

MC_Identify

Syntax:

```
bool AutoField()
```

Parameters

None

Returns

False, if the action is not applied at the page level. Otherwise, True.

Level

Page level.

Details

This action Identifies red HCFA-1500 or red UB04 forms.

This action must be placed in the rule after the SetMaxToleranceDistance, SetFormType, ReadDCOSetup, and SetWritePosFile actions.

Example:

```
SetMaxToleranceDistance(60)
SetFormType(0)
ReadDCOSetup(HFCA.xml, POS 1052)
AutoField()
```

Parent topic: [MC_Identify](#)

FindFields

Sets up a data file for the current page and provides field position information to the data file.

Member of namespace:

MC_Identify

Syntax:

```
bool FindFields()
```

Parameters

None

Returns

False, if the rule with this action is not bound to a page object of the document hierarchy or if a source page that is represented by the page object is not available. Otherwise, True.

Level

Page level.

Details

This action sets up a data file (.xml) for the current page and supplies the data field with field position information.

This action is usually part of a Medical Claims ID_PageFix rule.

Example:

```
SetMaxToleranceDistance (60)
SetFormType (0)
ReadDCOSetup (HCFA.xml, POS 1052)
FindFields`()
```

Parent topic: [MC_Identify](#)

ReadDCOSetup

Designates the file name of the document hierarchy.

Member of namespace:

MC_Identify

Syntax:

```
bool ReadDCOSetup (StrParam)
```

Parameters

A comma-separated string value that is made up of a smart parameter that designates the file name of the document hierarchy followed by a variable name that indicates the position of the fingerprint. The file name is usually the application ID with an .xml extension. For example, HCFA.xml is name of document hierarchy file of the HCFA application. POS 1052 specifies a previously assembled HCFA-1500 fingerprint with details of the form's Field IDs, locations, and data types. For use with the Application Service, the syntax of the first parameter changes to include the smart parameter that points to the Setup DCO. For example, it uses @APPPATH(setupdco) instead of HCFA.xml.

Returns

False, if the document hierarchy is not found. Otherwise, True.

Level

All levels, but usually the Batch level.

Details

Example:

```
ReadDCOSetup (HFCA.xml, POS 1052)
```

For use with the Application Manager with the @APPPATH smart parameter to locate the Setup DCO:

```
ReadDCOSetup (@APPPATH (setupdco) , POS 1052)
```

This action is used with almost every other MC_Identify action.

Parent topic: [MC_Identify](#)

ReadPageSetup

Designates the file name and path of the document hierarchy.

Member of namespace:

MC_Identify

Syntax:

```
bool ReadPageSetup (string DCOSetupPath, string FPPosition, string PageType)
```

Parameters

string DCOSetupPath

string FPPosition

string PageType

Parameters

1. PageType: A smart parameter enabled argument that designates the file name and path of the document hierarchy.
2. FPPosition: The fingerprint position variable, usually Pos followed by the ID of the desired fingerprint.
3. PageType: The page type of the page to identify, usually PClaim or IClaim.

Returns

False, if the document hierarchy file is not found. Otherwise, True.

Level

All levels.

Details

Example:

```
ReadPageSetup (@APPPATH (setupdco) , POS 1059, PClaim)
```

This action is used with almost every other MC_Identify action.

Parent topic: [MC_Identify](#)

SetFormType

Set the value of the Form Type that is used by the AutoField action.

Member of namespace:

MC_Identify

Syntax:

```
bool SetFormType (StrParam)
```

Parameters

String value that indicates the Form Type:

- For HCFA-1500 or CMS-1500, use 0 or hcfa.
- For UB-04, use 2 or ub04.

Returns

False, if the parameter is invalid. Otherwise, True.

Level

All levels.

Details

This action sets the Form Type value that is used by the AutoField action.

Example:

```
SetMaxToleranceDistance (60)
SetFormType (0)
ReadDCOSetup (HFCA.xml, POS 1052)
SetWritePosFile (True)
AutoField ()
```

Parent topic: [MC_Identify](#)

SetMaxTolerantDistance

Set the tolerance level that the AutoField action uses to match HCFA-1500 or red UB04 forms.

Member of namespace:

MC_Identify

Syntax:

```
bool SetMaxTolerantDistance (StrParam)
```

Parameters

The Maximum Tolerance Distance: an integer from 1 for the lowest tolerance to 100 for the highest tolerance.

Returns

False, if the parameter is not an integer from 1 to 100. Otherwise, True.

Level

All levels.

Details

This action sets the tolerance level that is used by the AutoField action to match HCFA-1500 or red UB04 forms based on the form that is specified in the SetFormType action.

Example:

```
SetMaxToleranceDistance (60)
SetFormType (0)
ReadDCOSetup (HFCA.xml, POS 1052)
AutoField ()
```

Parent topic: [MC_Identify](#)

MC_Validation

Use the MC_Validation actions to validate medical claim form information.

The MC_Identify actions validates the information in the medical claim forms that are processing in the batch.

- [AddCenturyTo2YearDigit](#)
Converts two-digit Year values to four-digit Year values.
- [AddToDetailErrorMsg](#)
Adds the specified value to the existing value for the page variable *ErrorMessage*.
- [AddToErrorMsg](#)
Adds the specified value to the existing value for the page variable *ErrorMessage*.
- [CalculateHCFALineCharges](#)
Calculates charges for HCFA service lines.
- [CalculateUBLLineCharges](#)
Calculates charges for UB service lines.
- [CheckDocID](#)
Checks the document IDs and updates them to the proper format.
- [ClearErrorMsg](#)
Clears the value of the page variable *ErrorMessage*.
- [CommonParseAddress](#)
Parses the addresses in the HCFA and UB04 fields into appropriate subfields.
- [CommonValAddress](#)
Validates the address values first name, last name, street, city, state, zip code, and phone number.
- [ConvertHyphen](#)
Removes spaces, commas, hyphens, and invalid characters.

- [FilterPID](#)
Filters the qualifier from the attending physician for UB04 claims.
- [FormatFieldLengths](#)
Truncates the length of the field and sets the last character of the field to low confidence.
- [InheritSnippets](#)
Assigns the snippet position information of the current Field object to the Field objects that are specified in the parameter.
- [MC_ReadZones](#)
Adjusts the autofield that is based on the OMR field zone positions on the calling page.
- [Parse31aPhSig](#)
Parses the 31aPhSig field of the HFCA application.
- [Parse58ainsnm](#)
Parses the 58ainsnm field of the UB04 application.
- [Parse58binsnm](#)
Parses the 58binsnm field of the UB04 application.
- [Parse58cinsnm](#)
parses the 58cinsnm field of the UB04 application.
- [ParseConditionCodes](#)
Detects and parses Condition Code data elements from the calling field value.
- [ParseEPSDT](#)
Detects and parses an ESDT Reason Code from the calling field value.
- [ParseLastFirstIniNames](#)
Parses the name information in the first line of an address superfield.
- [ParseNDC](#)
Detects and parses NDC data elements from the calling field value.
- [PopulateFromField](#)
Copies the value from field that is specified by the parameter into the current field.
- [SetConf](#)
Set a confidence string for a field.
- [SetOriginalTIF](#)
Replaces the current TIF file with the original TIF image that was previously renamed with a TI1 extension.
- [StripTrailingAlpha](#)
Removes all alpha characters from the captured value, except from the first character position.
- [TransformLI](#)
Assigns values to the fields in the lines of the Line Item Table.
- [UpdateCredentialList](#)
Updates the default list of abbreviations that are used by the parsing actions to extract credential strings from names.
- [ValidateNPI](#)
Validates the NPI value by evaluating the 10 digits in the value that use a modified LUHN checkdigit algorithm.
- [ValProcedureCode](#)
Validates the Procedure Code fields in a HCFA-1500 form.
- [ValRequiredGroup](#)
Checks that all of the fields in a designated group are filled with data.

Parent topic: [Global actions](#)

AddCenturyTo2YearDigit

Converts two-digit Year values to four-digit Year values.

Member of namespace:

MC_Validation

Syntax:

```
bool AddCenturyTo2YearDigit()
```

Parameters

None.

Returns

False, if the value is not a valid date in the mmddyy format or if the action is not applied at the Field level. Otherwise, True.

Level

Field level.

Details

This action converts two-digit Year values to four-digit Year values.

All dates are assumed to be before today's date with a format of mmddyy. If today is 051507 and this action is applied to a field with a value of 102295, the date is assumed to be 10221995.

Example:

```
AddCenturyTo2YearDigit()
```

Parent topic: [MC_Validation](#)

AddToDetailErrorMsg

Adds the specified value to the existing value for the page variable *ErrorMessage*.

Member of namespace:

MC_Validation

Syntax:

```
bool AddToDetailErrorMsg(StrParam)
```

Parameters

1. A smart parameter or regular string to add to the error message variable.
2. Optional comma separated second parameter to trigger the action to return True.

Returns

True, if the optional second comma separated parameter is used. Otherwise, False.

Level

Field level.

Details

This action adds the specified value to the existing value for the page variable *ErrorMessage*.

Example:

```
AddToDetailErrorMsg("Description cannot be blank")
```

Parent topic: [MC_Validation](#)

AddToErrorMsg

Adds the specified value to the existing value for the page variable *ErrorMessage*.

Member of namespace:

MC_Validation

Syntax:

```
bool AddToErrorMsg(StrParam)
```

Parameters

1. A smart parameter or regular string to add to the error message variable.
2. Optional comma separated second parameter to trigger the action to return True.

Returns

True, if the optional second comma separated parameter is used. Otherwise, False.

Level

Field level.

Details

This action adds the specified value to the existing value for the page variable *ErrorMessage*.

Example:

```
AddToErrorMsg("Invoice Number must be 60% numeric with a  
minimum length of 2.")
```

Parent topic: [MC_Validation](#)

CalculateHCFALineCharges

Calculates charges for HCFA service lines.

Member of namespace:

MC_Validation

Syntax:

```
bool CalculateHCFALineCharges ()
```

Parameters

None.

Returns

True, if the line charges equal the charges field. Otherwise, False.

Level

Field level.

Details

This action calculates charges for HCFA service lines.

Example:

```
CalculateHCFALineCharges ()
```

Parent topic: [MC_Validation](#)

CalculateUBLLineCharges

Calculates charges for UB service lines.

Member of namespace:

MC_Validation

Syntax:

```
bool CalculateUBLLineCharges ()
```

Parameters

None.

Returns

True, if the line charges equal the charges field. Otherwise, False.

Level

Field level.

Details

This action calculates charges for UB service lines.

Example:

```
CalculateUBLLineCharges ()
```

Parent topic: [MC_Validation](#)

CheckDocID

Checks the document IDs and updates them to the proper format.

Member of namespace:

MC_Validation

Syntax:

```
bool CheckDocID ()
```

Parameters

None.

Returns

True, if the docid is formatted without an error. Otherwise, False.

Level

Document level.

Details

This action checks the document IDs and updates them to the proper format.

Example:

```
CheckDocID ()
```

Parent topic: [MC_Validation](#)

ClearErrorMsg

Clears the value of the page variable *ErrorMessage*.

Member of namespace:

MC_Validation

Syntax:

```
bool ClearErrorMsg()
```

Parameters

None.

Returns

Always True.

Level

Page level.

Details

This action clears the value of the page variable *ErrorMessage*.

Example:

```
ClearErrorMsg()
```

Parent topic: [MC_Validation](#)

CommonParseAddress

Parses the addresses in the HCFA and UB04 fields into appropriate subfields.

Member of namespace:

MC_Validation

Syntax:

```
bool CommonParseAddress(StrParam)
```

Parameters

For Professional Claim, string value of:

1. HCField32Object: for parsing the Facility Address field (field 32).
2. HCField33Object: for parsing the Physician Address field (field 33).

For Institutional Claim, string value of:

1. UBField1Object: for parsing the Provider Address field (field 1).
2. UBField2Object: for parsing the Pay-To Address field (field 2).
3. UBField38Object: for parsing the Responsible Party Name and Address field (field 38).

Returns

False, if the parameter is invalid or if the action is not on the Page level. Otherwise, True.

Level

Page level.

Details

This action parses the addresses in the following fields into appropriate subfields:

- Professional Claims - Facility Address (field 32) or Physician Address (field 33)
- Institutional Claims - Provider Address (field 1), Pay-To Address (field 2), or Responsible Party Name and Address (field 38).

Example:

```
CommonParseAddress (HCField32Object)
```

Parent topic: [MC_Validation](#)

CommonValAddress

Validates the address values first name, last name, street, city, state, zip code, and phone number.

Member of namespace:

MC_Validation

Syntax:

```
bool CommonValAddress (StrString)
```

Parameters

Comma-delimited String that contains a list of name with addresses to be validated.

Returns

False, if the action is not run at the Page level. Otherwise, True.

Level

Page level.

Details

This action validates the following address values:

1. First Name: value can start with Ms, Mr, Miss, Dr salutations. The remaining values must be alphanumeric with no special characters. Punctuation is allowed only after the salutation.
2. Last Name: same requirements as the first name.
3. Street: alphanumeric, upper or lower case. Can include punctuation and the # character.
4. City: characters from A to Z, upper or lower case, comma, period, space, and the & character.
5. State: must be 2 alphanumeric characters.
6. Zip Code: must be between 5 and 9 characters. This value is checked against the State value above.

7. Phone Number: the area code is checked against the State and Zip Code values above.

Example:

```
CommonValAddress (Insured, 4InsFNam, 4InsLNam, 7|AddStr, 7|AddCity,  
7|AddSta, 7|AddZip)  
or  
CommonValAddress (Description, 12plname, 12pfname, 13paddr1, 13paddr2,  
13padcit, 13padsta, 13padzip)
```

Parent topic: [MC_Validation](#)

ConvertHyphen

Removes spaces, commas, hyphens, and invalid characters.

Member of namespace:

MC_Validation

Syntax:

```
bool ConvertHyphen ()
```

Parameters

None.

Returns

False, if the parameter is not called at the Field level. Otherwise, True.

Level

Field level.

Details

This action removes spaces, commas, hyphens, and invalid characters.

"1,2,3,4" becomes "1234", "1-2-3" becomes "123". Valid characters for this field are {1,2,3,4}.

Characters other than 1,2,3,4, space, commas, or hyphens will lower the field confidence level.

Example:

```
ConvertHyphen ()
```

Parent topic: [MC_Validation](#)

FilterPID

Filters the qualifier from the attending physician for UB04 claims.

Member of namespace:

MC_Validation

Syntax:

```
bool FilterPID(StrParam)
```

Parameters

The name of the field to filter.

Returns

False, if not called at the Field level. Otherwise, True.

Level

Field level.

Details

This action filters the qualifier from the attending physician for UB04 claims.

Example:

```
FilterPID(76apqual)
```

Parent topic: [MC_Validation](#)

FormatFieldLengths

Truncates the length of the field and sets the last character of the field to low confidence.

Member of namespace:

MC_Validation

Syntax:

```
bool FormatFieldLengths()
```

Parameters

None.

Returns

Always True.

Level

Page level.

Details

This action truncates the length of the field and sets the last character of the field to low confidence.

Example:

```
FormatFieldLengths()
```

Parent topic: [MC_Validation](#)

InheritSnippets

Assigns the snippet position information of the current Field object to the Field objects that are specified in the parameter.

Member of namespace:

MC_Identify

Syntax:

```
bool InheritSnippets(StrParam)
```

Parameters

The names of the fields that will inherit the same snippet information as the current Field object.

For example, 2paLname, 2PaFname, aPaMInit.

Returns

False, if the action is not called at the Field level or if a parameter is incorrect. Otherwise, True.

Level

Field level.

Details

This action assigns the snippet position information of the current Field object to the Field objects that are specified in the parameter.

Example:

```
InheritSnippets(2paLname, 2PaFname, aPaMInit)
```

Parent topic: [MC_Validation](#)

MC_ReadZones

Adjusts the autofield that is based on the OMR field zone positions on the calling page.

Member of namespace:

MC_Validation

Syntax:

```
bool MC_ReadZones ()
```

Parameters

None.

Returns

False, if the action is not called at the Page level. Otherwise, True.

Level

Page level.

Details

This action adjusts the autofield that is based on the OMR field zone positions on the calling page. Important: This action handles Autofield-based OMR zone detection for Medical Claims application. This action is not compatible with standard rules-based OMR zone detection procedures.

Example:

```
MC_ReadZones ()
```

Parent topic: [MC_Validation](#)

Parse31aPhSig

Parses the 31aPhSig field of the HFCA application.

Member of namespace:

MC_Validation

Syntax:

```
bool Parse31aPhSig ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

This action parses the 31aPhSig field of the HFCA application.

Example:

```
Parse31aPhSig()
```

Parent topic: [MC_Validation](#)

Parse58ainsnm

Parses the 58ainsnm field of the UB04 application.

Member of namespace:

MC_Validation

Syntax:

```
bool Parse58ainsnm()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

This action parses the 58ainsnm field of the UB04 application.

Example:

```
Parse58ainsnm()
```

Parent topic: [MC_Validation](#)

Parse58binsnm

Parses the 58binsnm field of the UB04 application.

Member of namespace:

MC_Validation

Syntax:

```
bool Parse58binsnm()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

This action parses the 58binsnm field of the UB04 application.

Example:

```
Parse58binsnm()
```

Parent topic: [MC_Validation](#)

Parse58cinsnm

parses the 58cinsnm field of the UB04 application.

Member of namespace:

MC_Validation

Syntax:

```
bool Parse58cinsnm()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

This action parses the 58cinsnm field of the UB04 application.

Example:

Parse58cinsnm()

Parent topic: [MC_Validation](#)

ParseConditionCodes

Detects and parses Condition Code data elements from the calling field value.

Member of namespace:

MC_Validation

Syntax:

```
bool ParseConditionCodes()
```

Parameters

None.

Returns

False, if the action generates an error or an invalid code is detected. Otherwise, True.

Level

Field level that contains Condition Code data to detect and parse to fields CCode 1 through CCode4.

Details

This action detects and parses Condition Code data elements from the calling field value.

Valid codes are: AA, AB, AC, AD, AE, AF, AG, AH, AI, W2, W3, W4, or W5.

Example:

```
ParseConditionCodes()
```

Parent topic: [MC_Validation](#)

ParseEPSDT

Detects and parses an ESDT Reason Code from the calling field value.

Member of namespace:

MC_Validation

Syntax:

```
bool ParseEPSDT(string EPSDTCode)
```

Parameters

A Smart Parameter enabled value that represents the target field where the parsed data is to be saved.

Returns

False, if the action generates an error. Otherwise, True.

Level

Field level that contains the Reason Code data.

Details

This action detects and parses an ESDT Reason Code from the calling field value. Parsing looks for "AV", "S2", "ST", or "NU" as a single word at the end of the field value.

Example:

```
ParseEPSDT ("..\EPSDTCode")
```

Parent topic: [MC_Validation](#)

ParseLastFirstIniNames

Parses the name information in the first line of an address superfield.

Member of namespace:

MC_Validation

Syntax:

```
bool ParseLastFirstIniNames (StrParam)
```

Parameters

These comma separated parameters:

1. The name of the Last Name Field object.
2. The name of the First Name Field object.
3. The name of the Middle Name or Middle Initial object.
4. The name of the Credential Field object.
5. The name of the Suffix Field object.

Returns

False, if the parameter values are invalid. Otherwise, True.

Level

Field level.

Details

This action parses the name information in the first line of an address superfield.

The action parses the value of the full name into the Last, First, and MiddleName/Initial fields that are specified by the parameter. In the absence of any explicit pattern, such as a punctuation mark or a middle initial, parsing defaults to First Middle Last. A parameter value of -1 in the argument changes this default in the absence of any explicit pattern to Last First.

Example:

For form fields where the instructions specify First Middle Last:

```
ParseLastFirstIniNames (8plname, 8pfname, 8pminit)
```

For form fields where the instructions specify Last First Middle:

```
ParseLastFirstIniNames (17RelLNam, 17RelFNam, 17RelMini,  
17RelCred, 17RelSufx, -1)
```

Parent topic: [MC_Validation](#)

ParseNDC

Detects and parses NDC data elements from the calling field value.

Member of namespace:

MC_Validation

Syntax:

```
bool ParseNDC(string NDCField, string TypeField, string QuantityField)
```

Parameters

Three Smart Parameters that represent the target path from the calling object to the field's parsed data that is to be saved.

Returns

False, if the action generates an error. Otherwise, True.

Level

Field level that contains NDC data to detect and parse.

Details

This action detects and parses NDC data elements from the calling field value. NDC value parsing looks for "N4" followed by 11 numbers. The NDC Type and Quantity parameters look for "F2", "GR", "ML", or "UN" followed by 1 to 9 numbers.

Example:

```
ParseNDC ("..\NDC", "..\NDCType", "..\NDCQty")
```

Parent topic: [MC_Validation](#)

PopulateFromField

Copies the value from field that is specified by the parameter into the current field.

Member of namespace:

MC_Validation

Syntax:

```
bool PopulateFromField(StrParam)
```

Parameters

The name of the field whose value is to be assigned to the current field

Returns

False, if the parameter is invalid or if the action is not applied at the Field level. Otherwise, True.

Level

Field level.

Details

This action copies the value from field that is specified by the parameter into the current field.

Example:

```
PopulateFromField(24aDtFr1)
```

Parent topic: [MC_Validation](#)

SetConf

Set a confidence string for a field.

Member of namespace:

MC_Validation

Syntax:

```
bool SetConf(StrParam)
```

Parameters

The value of the new confidence for each character in the field, 0 to 9.

Returns

Always True.

Level

Field level.

Details

This action sets a confidence string for a field.

Example:

```
SetConf(9)
```

This example sets the confidence for each character in the field to 9, which is the highest confidence.

Parent topic: [MC_Validation](#)

SetOriginalTIF

Replaces the current TIF file with the original TIF image that was previously renamed with a TI1 extension.

Member of namespace:

MC_Validation

Syntax:

```
bool SetOriginalTIF(StrParam)
```

Parameters

The extension of the original image file.

Returns

False, if the original image does not exist. Otherwise, True.

Level

Page level.

Details

This action replaces the current TIF file with the original TIF image that was previously renamed with a TI1 extension. It is assumed that the original file name was copied to a file name that uses a different extension for safe keeping.

Example:

```
SetOriginalTIF(TI1)
```

This example replaces the current TIF file with the original TIF image that was previously renamed with a TIF extension.

Parent topic: [MC_Validation](#)

StripTrailingAlpha

Removes all alpha characters from the captured value, except from the first character position.

Member of namespace:

MC_Validation

Syntax:

```
bool StripTrailingAlpha()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

This action removes all alpha characters from the captured value, except from the first character position.

Example:

```
StripTrailingAlpha()
```

Parent topic: [MC_Validation](#)

TransformLI

Assigns values to the fields in the lines of the Line Item Table.

Member of namespace:

MC_Validation

Syntax:

```
bool TransformLI()
```

Parameters

None.

Returns

True if the assignment of values is successful for all of the fields in all of the lines of the Line Item Table.

False, if the transfer of values for one or more of the fields fails.

Level

Document or Page level.

Details

Each line in the table is remapped into a repeating set of fields. These fields have a parent field with a unique ID. All of the data in a source field is retained in the new target field, with the exception of the new fields type and ID. The linear field structure is replaced with a structure that is based on a parent field in the claim page called DETAILS.

Each set of fields in the row of table data is then placed in a LINEITEM type field. The ID of the field is patterned on the index of its insertion in the DETAILS field. The first Line Item field is called LINEITEM1, the second is LINEITEM2, and so on. Each Line Item field contains an identical set of Field Type and ID as outlined in the following values list.

For example, the HCFA Line Item Table has 12 24nDtfr fields. During processing, the TransformLI action assigns recognized values for these fields to a Field object named DateFrom that is a child of the Line Item Table Details new parent Field object. So, 12 recognized values can be assigned to the Field object DateFrom.

Values that are associated with rows of the HCFA-1500 table are assigned to the Field types where n is value from 1 to 12:

- 24aDtn = DateFrom
- 24aDtTon = DateThru
- 24bPlacn = PlaceOfService
- 24cTypen = TypeOfService
- 24cEMG_n = EMG_C
- 24dCPT_n = CPT_Code
- 24dModin = Modifiers
- 24eDiagn = DiagPointer
- 24fChgsn = Charges
- 24gDaysn = Days_Units
- 24hEPDn = EPD
- 24iQualn = Qualifier
- 24jReflDn = ReferenceId
- 24jEMG_n = EMG_I
- 24jCOBn = COB
- 24kLocn = LocalUse
- 24lInfn = Info

Values that are associated with the rows of a UB04 table are assigned to the child Field objects where n is a value from a to z:

- 42nrevcd = RevCode
- 43ndscrt = Description
- 44nhcpcs = HCPCS

- 44nMod = Modifiers
- 45nsrvdt = ServiceDate
- 46nsrvun = Units
- 47nttchg = Charges
- 48nncchg = NotCovered
- 49n = LocalUse

Note: This action converts all of the pages in a document if it called from a Document object. The expected page types are HCFA, 1500 and UB04_Page.

Parent topic: [MC_Validation](#)

UpdateCredentialList

Updates the default list of abbreviations that are used by the parsing actions to extract credential strings from names.

Member of namespace:

MC_Validation

Syntax:

```
bool UpdateCredentialList(string sCredential,string AddRemove)
```

Parameters

string sCredential

string AddRemove

Parameters

A two parameter specification of a list of credential abbreviations and a Add or Remove from the list indicator.

sCredential is a comma separated list of credential abbreviations. Smart parameters are supported.

AddRemove indicator is an Add or Remove Condition from the list command. Defaults to Add. The keywords for setting the remove from the list mode are: "OFF", "REMOVE", "0", "FALSE", "-1", "NO", and "DELETE".

Returns

False, if the action generates an error. Otherwise, True.

Level

All levels.

Details

This action updates the default list of abbreviations that are used by the following name parsing actions to extract credential strings from names: ParseLastFirstIniNames(), Parse58ainsnm(), Parse58binsnm(), Parse58cinsnm(), Parse82name(), Parse83aname(), Parse83bname(), Parse31aPhSig(),

Example:

```
UpdateCredentialList (MD, Add)
```

Parent topic: [MC_Validation](#)

ValidateNPI

Validates the NPI value by evaluating the 10 digits in the value that use a modified LUHN checkdigit algorithm.

Member of namespace:

MC_Validation

Syntax:

```
bool ValidateNPI ()
```

Parameters

None.

Returns

True, if the field contains a valid NPI value. Otherwise, False.

Level

Field level.

Details

This action validates the NPI value by evaluating the 10 digits in the value that use a modified LUHN checkdigit algorithm.

Example:

```
ValidateNPI ()
```

Parent topic: [MC_Validation](#)

ValProcedureCode

Validates the Procedure Code fields in a HCFA-1500 form.

Member of namespace:

MC_Validation

Syntax:

```
bool ValProcedureCode ()
```

Parameters

None.

Returns

False, if the action is not applied at the Field level or if the Procedure Code is invalid. Otherwise, True.

Level

Field level.

Details

This action validates the Procedure Code fields in a HCFA-1500 form.

Example:

```
ValProcedureCode ()
```

Parent topic: [MC_Validation](#)

ValRequiredGroup

Checks that all of the fields in a designated group are filled with data.

Member of namespace:

MC_Validation

Syntax:

```
bool ValRequiredGroup (StrParam)
```

Parameters

The names of fields in the group.

Returns

False, if the parameters are invalid or if any of the parameter fields do not contain data. Otherwise, True.

Level

Field level.

Details

This action checks that all of the fields in a designated group are filled with data.

Example:

```
ValRequiredGroup (24aDtFr1,24aDtTo1,24adCPT_1,24fChg1,24gdays1)
```

mvscan actions

Use mvscan actions in place of the vscan actions to create batches from files that are stored on your local disk. The mvscan actions can handle large numbers of files that are stored in the input folder.

The mvscan actions can run simultaneously on multiple stations, in multiple threads. These actions can read from a single folder or multiple folders to provide high availability and throughput. The mvscan actions can also read metadata and include it in the batch with the input files. Refer to the [set_metadata_types](#) action for details.

Note: If a batch aborts, and the files are not moved; for example, if there is no problem folder, then the batch aborts repeatedly, and the Rulerunner does not process normally.

- [mv_retain_folder](#)
Indicates whether to retain the directory tree structure of a file's input folder when the file is placed in the copy or problem folder.
- [scan](#)
Polls the folder that is specified by the [set_folder](#) action and scans the images into a batch folder.
- [set_abort_time](#)
Specifies the length of time in seconds to delay processing if the scan action encounters an unrecoverable error and sets the batch status to aborted.
- [set_copy_folder](#)
Optional, sets a folder to contain a copy of each ingested file.
- [set_delete_empty_folders](#)
Determines whether to delete the empty subdirectories.
- [set_folder](#)
Specifies the folder or folders in which to search for the image files that you want to scan.
- [set_image_validation](#)
Causes the Scan action to validate that the specified input files are all valid image files. Invalid files are moved to the problem folder.
- [set_max_docs](#)
Specifies maximum number of pages in each batch when used with the [set_types](#) action.
- [set_metadata_types](#)
Specifies the input file name extensions of XML files that contain the names of the files to ingest and the metadata to include in the batch for each file.
- [set_min_age](#)
Specifies the minimum time in seconds to wait after the file was modified before ingesting the file.
- [set_move_wait_time](#)
Specifies the time to wait for a source file to be deleted after it is moved from to the batch folder.
- [set_multipage_burst](#)
Forces multipage TIFF images to be split into single pages during ingestion.
- [set_problem_folder](#)
Sets the folder into which the files that cannot be ingested are placed.
- [set_sort_method](#)
Selects method to use for sorting files for ingestion.
- [set_tree_mode](#)
Determines whether subdirectories are included in the scan for files to ingest.
- [set_types](#)
Specifies the extensions of the file types to ingest in a comma-separated list of file extensions.
- [set_wait_time](#)
For batch creation purposes, specifies the maximum interval that the scan action waits for the arrival of more input files to add to a batch.

mv_retain_folder

Indicates whether to retain the directory tree structure of a file's input folder when the file is placed in the copy or problem folder.

Member of namespace

mvscan

Syntax

```
mv_retain_folder(bool bRetainFolders)
```

Parameters

bRetainFolders

Whether to retain the directory tree structure of a file's input folder. Here are the possible values:

True	Within the copy and problem folders, the scan action places files in a directory structure that partly replicates the file's original input folder. Also, the scan action creates several variables that indicate various folder and file paths. For further explanation, see the Details section.
False	Input files are placed directly in the copy or problem folders.

By default, if you do not call this action, the directory tree structure of a file's input folder is not retained.

Returns

Always True.

Level

Batch level.

Details

Use this action to retain a file's input folder structure in the copy or problem folder. For an explanation of the specific manner in which folder structure is retained, see the example.

This action sets the following variables:

Variable	Description	Example
ScanSrcInputFolder	The full path of the input folder for this file. For more information, see ScanSrcInputFolder .	c:\shared\group1
ScanSrcSubFolder	The relative path of the input folder in which the file was found, which includes the root input folder name but not the root folder path. For more information, see ScanSrcSubFolder .	\shared\group1\batch1

Variable	Description	Example
ScanSrcFile Name	The original filename without the path. For more information, see ScanSrcFileName .	invoice_0001.tif

You can see the values of these variables in the VScan.xml file. For information about this file, see [Examining the files in the runtime batch folder](#).

Example:

```
set_folder("c:\shared\group1|c:\shared\group2|d:\shared\group3")
set_copy_folder("InProgress")
set_problem_folder("Error")
mv_retain_folder(True)
scan()
```

In this example, the set_folder action specifies three input folders. The following table shows the copy and problem folders for each input folder:

Input folder	Root input folder	Copy folder	Problem folder
c:\shared\group1	c:\shared	c:\shared\InProgress	c:\shared\Error
c:\shared\group2	c:\shared	c:\shared\InProgress	c:\shared\Error
d:\shared\group3	d:\shared	d:\shared\InProgress	d:\shared\Error

Within the copy or problem folder, files are placed in a directory structure that replicates the file's original input folder except for the root input folder. The following table shows some examples:

Input folder file placement	Copy folder placement	Problem folder placement
c:\shared\group1\batch1	c:\shared\InProgress\group1\batch1	c:\shared\Error\group1\batch1
d:\shared\group3\batch99	c:\shared\InProgress\group3\batch99	d:\shared\Error\group3\batch99

Parent topic: [mvscan actions](#)

scan

Polls the folder that is specified by the set_folder action and scans the images into a batch folder.

Member of namespace

mvscan

Syntax

```
bool scan ()
```

Parameters

None.

Returns

False, if the operation fails, and pauses before the return. Otherwise, True.

Action returns when timeout is reached, or the requested number of files is ingested.

Level

Batch level Open event only.

Details

Scans the source folder for files with the extensions that you want, then ingests them into a batch. Call one time for each batch. All options must be set before this action is called. If no files are present for ingestion, the batch is set to pending status, and the action returns immediately. The files in each batch folder are ingested by the order of the date the file was last modified.

Each input file generates one or more "pages" in the batch with the following page variables set:

- TYPE : Always set to "Other".
- IMAGEFILE : The file name within the batch, for example TM000002.tif.

Example:

```
set_folder("@APP_PATH(vscaimagedir)+@STRING(\mvscan folder)")
set_types("jpg,pdf,tif")
set_max_docs("2")
scan()
```

Parent topic: [mvscan actions](#)

set_abort_time

Specifies the length of time in seconds to delay processing if the scan action encounters an unrecoverable error and sets the batch status to aborted.

Member of namespace

mvscan

Syntax

```
bool set_abort_time (int nSecs)
```

Parameters

nSecs

Type: int

Time to wait before continuing when a serious error occurs.

Parameters

nSecs : Time to wait before continuing when a serious error occurs.

Returns

Always True.

Level

Batch level.

Details

The scan action waits the specified time before returning if a fatal error is detected that would cause the batch to abort. This action can be useful to prevent a large number of aborted batches due to an abort condition. For example, if the source folder should become unavailable for some time, the abort timeout will limit the number of aborted batches until the folder becomes available again.

This delay restricts the number of aborted batches if the problem persists.

If this action is not called, the default abort time value of 5 seconds is used.

Example:

```
set_abort_time("60")
scan()
```

Parent topic: [mvscan actions](#)

set_copy_folder

Optional, sets a folder to contain a copy of each ingested file.

Member of namespace

mvscan

Syntax

```
bool set_copy_folder (string folderpath)
```

Parameters

Smart parameters are supported.

folderpath

The folder, if any, to which the scan action copies files that it successfully ingested. Specify a full or relative folder path. If you specify a relative path, the path is relative to the root input folder. For information about the root input folder, see [set_folder](#).

Here are the possible values:

Same as input folder	For demonstration purposes, you might want to set the copy folder to be the same as the input folder. In this case, ingested files are left in place.
Empty string	Ingested files are moved to the batch folder without making copies.
Any other folder	Ingested files are copied to the specified folder.

If you do not call this action, the scan action behaves by default the same as if you specified an empty string for the copy folder.

The location of the copy folder is affected by the `mv_retain_folder` action. For more information, see [mv_retain_folder](#).

Returns

Always True.

Level

Batch level.

Details

If the specified folder does not exist, the scan action generates an error.

This action must be called before `scan()` to take effect.

Example:

```
set_folder("@APPPATH(vscanimagedir)+@STRING(\input folder)")
set_copy_folder(@APPPATH(vscanimagedir)+@STRING(\copy folder))
scan()
```

Parent topic: [mvscan actions](#)

set_delete_empty_folders

Determines whether to delete the empty subdirectories.

Member of namespace

mvscan

Syntax

```
bool set_delete_empty_folders (bool bParam)
```

Parameters

bParam
Type: bool

Parameters

bParam : A Boolean value that enables or disables deleting sub-folders if they are empty. The root folder specified for ingestion is never deleted. Subfolders are only deleted if both this setting and tree mode are enabled.

True: Delete subfolders of the main ingestion folder if they are empty. This is the default if the action is not called.

False: Leave empty subfolders intact.

Details

If tree mode is not enabled, this setting has no effect. For this action to take effect, it must be called before the Scan action.

Example:

```
set_types("tif")
set_min_age("10")
set_tree_mode(True)
set_delete_empty_folders(False)
scan()
```

Parent topic: [mvscan actions](#)

set_folder

Specifies the folder or folders in which to search for the image files that you want to scan.

Member of namespace

mvscan

Syntax

```
bool set_folder (string folderpath)
```

Parameters

Smart parameters are supported.

folderpath

One or more input folders that are separated with a vertical bar (“|”). Specify the full path for each folder.

The scan action polls the specified input folders for files, which are incorporated into batches for processing.

Returns

Always True.

Level

Batch level.

Details

Use this action to specify the input folders for the scan action. To indicate that the subfolders of the input folders are also to be polled by the scan action, use the `set_tree_mode` action.

For any particular input folder, the root input folder is the parent folder of the input folder. For example, if the input folder is `c:\shared\group1`, the input root folder is `c:\shared`. The root input folder might be significant for other actions when you specify relative folder paths.

A batch contains files from one input folder only.

This action must be called before `scan()`.

Example:

```
set_folder("@APPPATH(vscanimagedir)+@STRING(\input folder)")
scan()

set_folder("c:\shared\group1|c:\shared\group2|d:\shared\group3")
scan()
```

Parent topic: [mvscan actions](#)

Related reference:

[scan](#)

[set_tree_mode](#)

set_image_validation

Causes the Scan action to validate that the specified input files are all valid image files. Invalid files are moved to the problem folder.

Member of namespace

mvscan

Syntax

```
bool set_image_validation (bool bValidate)
```

Parameters

bValidate
Type: bool

Parameters

bValidate : A Boolean value that enables or disables validating image files during ingestion.

True: Check to confirm that each file ingested contains a valid TIFF or JPG image.

False: Copy input files regardless of type or contents. This is the default if the action is not called.

Returns

Always True.

Level

Batch level.

Details

Enables testing that each file ingested is a valid image file. If enabled, converts black and white TIF files to G4 compression. This action should not be called against invalid file types such as PDFs due to incompatibility.

For this action to take effect, it must be called before the Scan action.

Example:

```
set_types("tif")
set_image_validation("True")
scan()
```

Parent topic: [mvscan actions](#)

set_max_docs

Specifies maximum number of pages in each batch when used with the set_types action.

Member of namespace

mvscan

Syntax

```
bool set_max_docs (int nDocs)
```

Parameters

nDocs

Type: int

Number of documents in a batch. Default 100.

Parameters

nDocs : An integer value which specifies the maximum number of input files to place into a batch.

If Metadata files are used, set_metadata_types() is called instead of set_types(), this specifies the maximum number of metadata files to ingest. All files referenced in each metadata file are included in the batch, there is no limit on the number of pages in the batch in this case. The default value is 100.

Returns

Always True.

Level

Batch level.

Details

For this action to take effect, it must be called before the Scan action.

Example:

```
set_tree_mode(false)
set_max_docs("1")
scan()
```

Parent topic: [mvscan actions](#)

set_metadata_types

Specifies the input file name extensions of XML files that contain the names of the files to ingest and the metadata to include in the batch for each file.

Member of namespace

mvscan

Syntax

```
bool set_metadata_types (string extensions)
```

Parameters

extensions

Type: string

Metadata image file extensions. If specified, XML metadata files (trigger files) control ingestion of files into batches, along with associated metadata. See more documentation for syntax and usage of metadata trigger files. If specified, these extensions override any prior call to `set_types()`.

Parameters

String value of extensions of input files that contain pointers to files to be ingested along with metadata to be included in the batch for each file. For multiple types, separate each one with a comma. Any files with extensions not in this list are ignored.

It is optional to include a period before each extension. The parameter is not case-sensitive.

Returns

Always True.

Level

Batch level Open event only.

Details

The format of the metadata file is documented here. This action overrides any previous call to `set_types()`, and forces the action to read properly formatted XML from the metadata trigger file. Then, the action determines which files get ingested, rather than ingesting individual files into the batch. Other settings such as `set_min_age()` are applied to the metadata files.

You must set up the XML file in the correct format to use the `set_metadata_type` action to scan metadata files into batches.

The following example illustrates a sample XML file:

```
<input>
  <item name="Invoice1" vendor_number="TS265329"
    vendor_name="Busy Car Repair" invoice_number="28100">
    <filename>C:\Datacap\APT\images\Input\
      Invoice_0001S1.tif</filename>
```

```

        <filename>C:\Datacap\APT\images\Input\
        Invoice_0002S1.tif</filename>
    </item>
    <item name="Invoice2" vendor_number="TS23785354"
        vendor_name="Trucking Co." invoice_number="876-3456">
        <filename>C:\Datacap\APT\images\Input\
        Invoice_0003S1.tif</filename>
        <filename>C:\Datacap\APT\images\Input\
        Invoice_0004S1.tif</filename>
    </item>
</input>

```

The XML file must follow this format.

1. The tag name of the XML root node must be `<input>`.
2. The `<item>` nodes represent a group of input files or images that share metadata.
3. The `<item>` nodes must contain at least one attribute, but can have a number of attributes. In this example, the `<item>` nodes contain the attributes `name=`, `vendor_number=`, `vendor_name=`, `invoice_number=`.
4. The `<item>` attribute names are entered in the item as text that follows the `=`.
5. These name values are prepended with `Meta_Item_` when they are written to the DCO to define the metadata variable names that are placed on each page. The attribute values populate the variable values.
6. The inner most nodes must be `<filename>` and contain items that represent single input files.
7. A `<filename>` can be either a fully qualified path, for example `\\server\folder\filename`, or a partial path relative to the location of the metadata file such as `folder\filename`. The number of file names in each repeating item is only limited to the practical limit of how many files you want to ingest in a single batch.
8. Each file is ingested as a Page in the batch, regardless of the file type or contents.

All of the items in a metadata file are ingested into the same batch. For example, if a metadata file contains 1 node with 50 file names, 50 files are ingested for that single metadata file. If the metadata file contains 5 nodes with 50 file names in each of them, then 250 files are ingested. The external system that creates the metafile must create all of the files to be ingested before it writes the metafile that refers to them.

Example:

```

set_metadata_types ("xml")
Scan ()

```

This sequence scans for metadata trigger files that end in ".xml".

Parent topic: [mvscan actions](#)

set_min_age

Specifies the minimum time in seconds to wait after the file was modified before ingesting the file.

Member of namespace

mvscan

Syntax

```
bool set_min_age (int nSecs)
```

Parameters

nSecs

The number of seconds since a file or group of files was last modified before the scan action ingests the file or files. As shown in the following table, the specific meaning of this parameter depends on the way that the scan action checks the minimum age of files:

Checking type	Parameter meaning	When
Individual	The number of seconds since a file was modified before the scan action ingests the file.	Non-tree mode or when the wait time is zero as set by the <code>set_wait_time</code> action
Collective	The number of seconds since any file was modified within a particular folder before the scan action ingests any file from that folder.	Tree mode except when the wait time is zero as set by the <code>set_wait_time</code> action

You set tree mode by calling `set_tree_mode`. For more information, see [set_tree_mode](#). For information about setting the wait time, see [set_wait_time](#).

By default, if you do not call this action, the default number of seconds that the scan action waits is zero. In this case, files are ingested as soon as they are present.

Returns

Always True.

Level

Batch level.

Details

This setting can be used to prevent premature ingestion of a file which may be incomplete. For example, a network scanner or multifunction device might create the image file and write contents over a period of time, rather than all at once. In that case ingesting the file immediately could cause errors, whereas waiting for 5 or 10 seconds would provide sufficient time for the file to be completed. If required, the appropriate value must be determined experimentally.

For this action to take effect, it must be called before the Scan action.

Example:

```
set_types("tif")
set_min_age("10")
scan()
```

Parent topic: [mvscan actions](#)

set_move_wait_time

Specifies the time to wait for a source file to be deleted after it is moved from to the batch folder.

Syntax

```
bool set_move_wait_time (int nSecs)
```

Parameters

int nSecs - Number of seconds to wait for the source file to be deleted before the action fails.

Returns

Always True.

Level

Batch level

Details

Over a slow network, and especially with large input files, moving files from the source folder to batch folder can take a significant length of time. The scan action checks that the source file was deleted after it was moved. If the source folder still exists, the scan action waits for a length of time periodically checking to see if the file was deleted.

If the folder still exists at the end of this time period, the copy in the batch folder is renamed with the extension .failed and it is not included in the batch. It is assumed that the original source file will be ingested by a subsequent scan task.

Example:

```
set_move_wait_time("120")
scan()
```

Parent topic: [mvscan actions](#)

set_multipage_burst

Forces multipage TIFF images to be split into single pages during ingestion.

Member of namespace

mvscan

Syntax

```
bool set_multipage_burst (bool bBurst)
```

Parameters

bBurst

Type: bool

Nonzero to force bursting of TIFF image files. Default = 0, no bursting.

Parameters

bBurst : A Boolean value that enables or disables bursting multipage image files during ingestion.

True: Any multipage TIFF file is separated into multiple pages in the batch, one per image.

False: One page is output per input file. This is the default if the action is not called.

Returns

Always True.

Level

Batch level.

Details

Enables splitting or bursting multipage source image files into one image per page in the batch. For this action to take effect, it must be called before the Scan action.

This action requires that `set_types()` be called with only one extension of TIF, TIFF, JPG or JPEG.

Example:

```
set_types(".tif")
set_multipage_burst(1)
scan()
```

If the scan action in this sequence encounters a multipage .tif file, it reads each one into the current batch as a separate image, thereby bursting the multipage file into individual images.

Parent topic: [mvscan actions](#)

set_problem_folder

Sets the folder into which the files that cannot be ingested are placed.

Member of namespace

mvscan

Syntax

```
bool set_problem_folder (string folderpath)
```

Parameters

folderpath

The folder to which the scan action moves files that it could not ingest. Specify a full or relative folder path. If you specify a relative path, the path is relative to the root input folder. For information about the root input folder, see [set_folder](#).

The location of the problem folder is affected by the `mv_retain_folder` action. For more information, see [mv_retain_folder](#).

Returns

Always True.

Level

Batch level.

Details

This action must be called before `scan()`. If this action is not called, and `scan` is unable to ingest a file, the batch is aborted. Files that are already ingested into the batch remain in the batch. The file that cannot be ingested might be in a locked state. Manual intervention is required.

Example:

```
set_folder("@APPPATH(vscanimagedir)+@STRING(\input folder)")
set_problem_folder(@APPPATH(vscanimagedir)+@STRING(\problem folder))
scan()
```

Parent topic: [mvscan actions](#)

set_sort_method

Selects method to use for sorting files for ingestion.

Member of namespace

mvscan

Syntax

```
bool set_sort_method (string method)
```

Parameters

method

Type: string

Sort method can be either DATE (default) or NAME. The parameter is not case sensitive.

Returns

Always True.

Level

Batch level Open event only.

Details

The scan action takes snapshots of the input folder, sorts the files by either date last modified (default) or by filename, and then tries to ingest the files in that order. This action allows you to override the default sort order.

Example:

```
set_sort_method("NAME")
Scan()
```

This sequence ingests files in alphabetical order.

Parent topic: [mvscan actions](#)

set_tree_mode

Determines whether subdirectories are included in the scan for files to ingest.

Member of namespace

mvscan

Syntax

```
bool set_tree_mode (bool bTreeFlag)
```

Parameters

bTreeFlag
Type: bool

Parameters

bTreeFlag : A Boolean value that enables or disables ingesting files from sub-folders within the top-level scan folder.

True: Search for files in subfolders to any depth under the scan folder. This is the default if the action is not called.

False: Ingest files that are located in the scan folder only. Ignore subfolders.

Details

If tree mode is enabled, each batch contains files from only one folder. Files from multiple folders are never ingested into the same batch.

For this action to take effect, it must be called before the Scan action.

Example:

```
set_types("tif")
set_tree_mode(True)
scan()
```

Parent topic: [mvscan actions](#)

set_types

Specifies the extensions of the file types to ingest in a comma-separated list of file extensions.

Member of namespace

mvscan

Syntax

```
bool set_types (string extensions)
```

Parameters

extensions

Type: string

Comma-separated list of file image file extensions to import

Parameters

String value of extension(s) of input files that should be ingested. For multiple types, separate each one with a comma. Any files with extensions not in this list are ignored. A smart parameter that evaluates to a file extension or list of extensions is accepted.

It is optional to include a period before each extension. The parameter is not case-sensitive.

Returns

Always True.

Level

Batch level Open event only.

Details

Uses the value of a file extension(s) to specify the type of files the task will scan.

This is an optional action: the task will scan .tif files by default. For this action to take effect, it must be called before the scan action.

This action overrides any previous call to `set_metadata_types()`, and causes files to be ingested without metadata.

Example:

```
set_types("tif,tiff,pdf")
scan()
```

This sequence ingests files ending with .tif, .tiff, and .pdf.

Parent topic: [mvscan actions](#)

set_wait_time

For batch creation purposes, specifies the maximum interval that the scan action waits for the arrival of more input files to add to a batch.

Important: This action can also affect the behavior of `set_min_age`. For more information, see [set_min_age](#).

Member of namespace

mvscan

Syntax

```
bool set_wait_time (int nSecs)
```

Parameters

nSecs

The maximum number of seconds to wait for files to complete a batch.

If you do not call this action, the default wait interval is 2 seconds.

Returns

Always True.

Level

Batch level.

Details

The maximum time to wait for input files to arrive, if at least one file was ingested, no more files are available, and the batch has not reached capacity. Ingestion of files into the batch stops when the wait limit is reached or when the maximum number of files per batch has been reached. If no files are available during the scan() action, and this time is reached, the scan action returns and the batch status remains pending.

Example:

```
set_wait_time("60")
scan()
```

Parent topic: [mvscan actions](#)

Maintenance Manager actions

The Maintenance Manager actions are divided into setup, batch processing, logging, and reporting categories.

The Maintenance Manager actions get connections to Datacap application databases and build query strings that run against these databases. These actions can also write logging information to Windows log files and send email messages that contain the log file. Maintenance Manager actions also write information to report tables in the Engine database for use by Datacap Report Viewer.

Embedded help is provided in Datacap Studio for all of the Maintenance Manager actions. To access the embedded help, select an action in the Actions Library tab and click information.

- [Application setup actions](#)
Use the Application setup actions to set up a connection from a Datacap application to Maintenance Manager.

- [Query setup actions](#)
Use the Query setup actions to build the query string that is run against the application databases that you connected to in the application setup.
- [Batch processing actions](#)
Use the Batch processing actions to run the SQL query and complete actions on the selected database records and, optionally, the corresponding batches.
- [Logging actions](#)
Use the Logging actions to write information to the Maintenance Manager and Windows log files and to send emails that contain the internal log file.
- [Reporting actions](#)
Use the Reporting actions to write information to the report tables in the Engine database for use by Datacap Report Viewer.

Parent topic: [Global actions](#)

Application setup actions

Use the Application setup actions to set up a connection from a Datacap application to Maintenance Manager.

The Application setup actions identify the name of the application and the Datacap Server to be used by Maintenance Manager. These actions also specify the login credentials, database information, and connection information that you need to connect a Datacap application to Maintenance Manager.

- [SetAdminDB](#)
Specifies the Administration database.
- [SetApplication](#)
Specifies the name of the Application used by Maintenance Manager.
- [SetEngineDB](#)
Specifies the Engine database.
- [SetPassword](#)
Action to set password to connect to the Server.
- [SetServer](#)
Specifies the name of the Datacap Server.
- [SetStation](#)
Action to set the station ID that is used to connect to the Server.
- [SetupDisconnectAll](#)
Disconnect from all Datacap servers.
- [SetupOpenApplication](#)
Creates connection to the application based on default settings.
- [SetupOpenApplicationEx](#)
Creates connection to the application based on the parameters provided.
- [SetUser](#)
Action to set a user name to login to the Server.

Parent topic: [Maintenance Manager actions](#)

SetAdminDB

Specifies the Administration database.

Member of namespace

Maintenance Manager

Syntax

```
bool SetAdminDB (string adminDB)
```

Parameters

adminDB
Type: string

Parameters

adminDB: Administration Database key in the Application Service. Smart parameters are supported.

Returns

Always True.

Level

Batch level.

Details

Obtains the Administrator database connection information from the application service. This is the default value used for the action SetupOpenApplication.

Example:

```
SetApplication("APT")
SetServer("Server 1")
SetAdminDB("*/tmadmin:cs")
SetEngineDB("*/tmengine:cs")
SetupOpenApplication("")
```

Parent topic: [Application setup actions](#)

Related reference:

[SetupOpenApplication](#)

[SetupOpenApplicationEx](#)

SetApplication

Specifies the name of the Application used by Maintenance Manager.

Member of namespace

Maintenance Manager

Syntax

```
bool SetApplication (string application)
```

Parameters

string application

Parameters

application: The application name. Smart parameters are supported.

Returns

False if the application name is missing. Otherwise, True.

Level

Batch level.

Details

Sets the name of the application, as defined in the application service, which are used by Maintenance Manager. The application determines the rules and databases that are used by subsequent actions. This will be the default value that is used for the SetupOpenApplication action.

Example:

```
SetApplication("APT")
SetServer("Server 1")
SetupOpenApplication("")
```

Parent topic: [Application setup actions](#)

Related reference:

[SetupOpenApplication](#)

[SetupOpenApplicationEx](#)

SetEngineDB

Specifies the Engine database.

Member of namespace

Maintenance Manager

Syntax

```
bool SetEngineDB (string engineDB)
```

Parameters

engineDB
Type: string

Parameters

engineDB: Engine Database key in the Application Service.

Returns

Always True.

Level

Batch level.

Details

Obtains the Engine database connection information from the application service. This will be the default value used for the action `SetupOpenApplication`.

Example:

```
SetApplication("APT")
SetServer("Server 1")
SetAdminDB("*/tmadmin:cs")
SetEngineDB("*/tmengine:cs")
SetUser("Admin")
```

Parent topic: [Application setup actions](#)

Related reference:

[SetupOpenApplication](#)

[SetupOpenApplicationEx](#)

SetPassword

Action to set password to connect to the Server.

Member of namespace

Maintenance Manager

Syntax

```
bool SetPassword (string password)
```

Parameters

String password

Parameters

password: Password. Smart parameters are supported.

Returns

Always True.

Level

Batch level.

Details

Specifies the password for the previously specified user ID. This will be the default value used for the SetupOpenApplication action.

When using LDAP or ADSI authentication, use the SetUser, SetStation, and SetApplication to login.

Example:

```
SetApplication("APT")
SetServer("Server 1")
SetAdminDB("*/tmadmin:cs")
SetEngineDB("*/tmengine:cs")
SetUser("Admin")
SetPassword("@APPVAR(values/adv/MyNENUPassword)")
SetupOpenApplication("")
```

This example uses the Smart Parameter @APPVAR to obtain the value of the password from the value name MyNENUPassword in the Custom Values tab of the Application Service Manager. The value name is configurable.

Parent topic: [Application setup actions](#)

Related reference:

[SetupOpenApplication](#)

[SetupOpenApplicationEx](#)

SetServer

Specifies the name of the Datacap Server.

Member of namespace

Maintenance Manager

Syntax

```
bool SetServer (string server)
```

Parameters

server

Type: string

Parameters

server: Server name. Smart parameters are supported.

Returns

False if a server name is not specified. Otherwise, True.

Level

Batch level.

Details

The name of the Datacap server as it is defined in the application service, which will be the server used by subsequent Maintenance Manager actions. This will be the default value used for the action SetupOpenApplication.

Example:

```
SetApplication("APT")
SetServer("Server 1")
SetupOpenApplication("")
```

Parent topic: [Application setup actions](#)

Related reference:

[SetupOpenApplication](#)

[SetupOpenApplicationEx](#)

SetStation

Action to set the station ID that is used to connect to the Server.

Member of namespace

Maintenance Manager

Syntax

```
bool SetStation (string station)
```

Parameters

string station

Parameters

station: Station ID. Smart parameters are supported.

Returns

Always True.

Level

Batch level.

Details

Specifies the station ID for login. This will be the default value that is used for the SetupOpenApplication action.

When using LDAP or ADSI authentication, use the SetUser, SetStation, and SetApplication to login.

Example:

```
SetApplication("APT")
SetServer("Server 1")
SetAdminDB("*/tmadmin:cs")
```

```
SetEngineDB ("*/tmengine:cs")
SetStation ("1")
SetupOpenApplication ("")
```

Parent topic: [Application setup actions](#)

Related reference:

[SetupOpenApplication](#)

[SetupOpenApplicationEx](#)

SetupDisconnectAll

Disconnect from all Datacap servers.

Member of namespace

Maintenance Manager

Syntax

```
bool SetupDisconnectAll ()
```

Parameters

None.

Returns

Always True.

Level

Any level.

Details

Closes the connections to all Datacap databases that were opened by Maintenance Manager actions. Some Maintenance Manager actions automatically close the connection to the database after a query has been performed.

Example:

```
SetupDisconnectAll ()
```

Parent topic: [Application setup actions](#)

SetupOpenApplication

Creates connection to the application based on default settings.

Member of namespace

Maintenance Manager

Syntax

```
bool SetupOpenApplication ()
```

Parameters

None.

Returns

True if the application exists in the application service the connection was successful. Otherwise, False.

Level

Batch level.

Details

A simplified version of `SetupOpenApplicationEx`. It is identical in operation with the difference that all of the default parameters are used. The default values can be set independently by individual actions, such as `SetApplication`, `SetServer`, etc. See `SetupOpenApplicationEx` for more information.

When using LDAP and LLDAP authentication the User, Password, and Station values must be blank.

Example:

```
SetApplication("APT")
SetServer("Server 1")
SetUser("user")
SetPassword("password")
SetAdminDB("* /tmadmin:cs")
SetEngineDB("* /tmengine:cs")
SetStation("1")
SetupOpenApplication()
```

Parent topic: [Application setup actions](#)

Related reference:

[SetupOpenApplicationEx](#)

[SetApplication](#)

[SetServer](#)

[SetAdminDB](#)

[SetEngineDB](#)

[SetUser](#)

[SetPassword](#)

[SetStation](#)

SetupOpenApplicationEx

Creates connection to the application based on the parameters provided.

Member of namespace

Maintenance Manager

Syntax

```
bool SetupOpenApplicationEx (string application, string server, string admin, string engine, bool debugFlag, string user, string password, string station)
```

Parameters

application
Type: string

server
Type: string

admin
Type: string

engine
Type: string

debugFlag
Type: bool

user
Type: string

password
Type: string

station
Type: string

Parameters

All parameters, except for debugFlag, support Smart parameters.

- application: Application name (Optional). Default value: the name of the application that is running this action.
- server: Server name (Optional). Default value: first available server in the application.
- admin: Administration database name (Optional). Default value: first available Admin DB.
- engine: Engine database name (Optional). Default value: first available Engine DB.
- debugFlag: Debug Flag (Optional). Default value: false.
- user: User name (Optional). Default value: current user credentials. Leave Blank for LDAP authentication.
- password: Password (Optional). Default value: current user credentials. Leave Blank for LDAP authentication.
- station: Station ID (Optional). Default value: Current station name. Leave Blank for LDAP authentication.

Returns

True, if the application exists in the application service the connection was successful. Otherwise, False.

Level

Batch level.

Details

Similar to SetupOpenApplication, this action connects Maintenance Manager to a specific application. The difference between the two actions is that here the default parameters can be overridden in one single action. These parameters determine what application is to be monitored by Maintenance Manager. This action, or

SetupOpenApplication, must be called first, then the Maintenance Manager actions can be used to run queries on the application and do required actions that are based on the results.

When you are using LDAP or ADSI authentication, use the SetUser, SetStation, and SetApplication actions instead of SetupOpenApplicationEx.

Example:

```
SetupOpenApplicationEX("Survey", "", "SurveyAdm.mdb", "SurveyEng.mdb", "False",  
"", "", "1")
```

Parent topic: [Application setup actions](#)

Related reference:

[SetupOpenApplication](#)

[SetApplication](#)

[SetServer](#)

[SetAdminDB](#)

[SetEngineDB](#)

[SetUser](#)

[SetPassword](#)

[SetStation](#)

SetUser

Action to set a user name to login to the Server.

Member of namespace

Maintenance Manager

Syntax

```
bool SetUser (string user)
```

Parameters

string user

Parameters

user: User name. Smart parameters are supported.

Returns

Always True.

Level

Batch level.

Details

Specifies the name of the user for the login to the Administrator and Engine databases. This will be the default value that is used for the SetupOpenApplication action.

When using LDAP or ADSI authentication, use the SetUser, SetStation, and SetApplication to login.

Example:

```
SetApplication("APT")
SetServer("Server 1")
SetAdminDB("*/tmadmin:cs")
SetEngineDB("*/tmengine:cs")
SetUser("Admin")
SetupOpenApplication("")
```

Parent topic: [Application setup actions](#)

Related reference:

[SetupOpenApplication](#)

[SetupOpenApplicationEx](#)

Query setup actions

Use the Query setup actions to build the query string that is run against the application databases that you connected to in the application setup.

Before, you can run a query against the application databases, you must build an SQL query string. When you run the actions in this category, Maintenance Manager appends the corresponding SQL code to the current query string. For example, running `QuerySetStatus("hold")` followed by `QuerySetOperator("admin")` generates the following query string.

```
Select * FROM JobMonitor WHERE queue.qu_status IN ('hold') AND qstats.qs_op IN ('admin')
```

- [QueryClear](#)
Clears the SQL query.
- [QuerySetAge](#)
Selects batches based on age using a date or number of seconds.
- [QuerySetBatchRange](#)
Sets range of batches for SQL query.
- [QuerySetBranch](#)
Sets the minimum number of children for the SQL query.
- [QuerySetDateFormat](#)
Sets custom Date format for SQL queries.
- [QuerySetDateRange](#)
Sets Date range for SQL query.
- [QuerySetDateTimeFormat](#)
Sets custom DateTime format for SQL queries.
- [QuerySetGeneric](#)
Builds an SQL query using the provided column name and value.
- [QuerySetJobID](#)
Sets the Job ID for the SQL query.
- [QuerySetOperator](#)
Sets the operator for the SQL query.
- [QuerySetPriority](#)
Sets the Priority for the SQL query.
- [QuerySetSeparator](#)
Sets SQL Date and Time Separator for SQL queries.

- [QuerySetStation](#)
Sets the station for the SQL query.
- [QuerySetStatus](#)
Sets the Task status for the SQL query.
- [QuerySetTaskID](#)
Sets the Task ID for the SQL query.

Parent topic: [Maintenance Manager actions](#)

QueryClear

Clears the SQL query.

Member of namespace

Maintenance Manager

Syntax

```
bool QueryClear ()
```

Parameters

None.

Returns

True if the query is cleared. Otherwise, False.

Level

Any level.

Details

Clears any SQL query that may be in memory. This action can be called to ensure that any Maintenance Manager query that you build is not building on any previous information.

Example:

```
QueryClear ("")
```

Parent topic: [Query setup actions](#)

Related reference:
[ProcessRunSqlQuery](#)

QuerySetAge

Selects batches based on age using a date or number of seconds.

Member of namespace

Syntax

```
bool QuerySetAge (string age, bool queryAgeStart)
```

Parameters

age
Type: string

queryAgeStart
Type: bool

Parameters

- age: All dates prior or within this date will be selected. Smart parameters are supported.
- queryAgeStart: If True, age will use the start time of the batch. If False, age will use the end time of the batch.

Returns

True if the query has been successfully set. It does not mean the query has been performed. Otherwise, False.

Level

Any level.

Details

Builds a query that selects batches based on the age of a batch. The age can be specified using a date or in seconds. The queryAgeStart parameter determines if the age is based on the start of the batch, the qs_start column, or based on the end of the batch, the qu_done column. If a date is specified, all batches will be selected based on the date. If a number is specified, all batches will be selected based on the number of seconds.

To control the age range use the exclamation point '!'. When the exclamation point is not specified, the age between now and the specified value will be selected. When the exclamation point is specified, the ages that are older than the value specified will be selected.

Example:

```
QuerySetAge ("300", "True")  
ProcessRunSqlQuery ("")
```

This example selects all batches started within last 300 seconds.

```
QuerySetAge ("!300", "true")  
ProcessRunSqlQuery ("")
```

This example selects all batches started prior to 300 seconds ago. Use the same pattern when using dates.

Parent topic: [Query setup actions](#)

Related reference:

[ProcessRunSqlQuery](#)

QuerySetBatchRange

Sets range of batches for SQL query.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetBatchRange (string start, string end)
```

Parameters

start

Type: string

end

Type: string

Parameters

- start: BatchID of the first batch in the range. Default value 00000000.000".
- end: BatchID of the last batch in the range. Default value zzzzzzzz.zzz".

Returns

True if the query has been successfully set. It does not mean the query has been performed. Otherwise, False.

Level

Any level.

Details

Allows selecting batch range based on the BatchID.

Example:

```
QuerySetBatchRange ("20110059.001", "20110059.010")  
InjectBatches ("20110059.001")
```

Parent topic: [Query setup actions](#)

Related reference:

[ProcessInjectBatches](#)

QuerySetBranch

Sets the minimum number of children for the SQL query.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetBranch (int children)
```

Parameters

children
Type: int

Parameters

children: The minimum number of children for the batch.

Returns

True if the query has been successfully set. It does not mean the query has been performed. Otherwise, False.

Level

Any level.

Details

Selects all of the batches that have at least the minimum number of specified children.

Example:

```
QuerySetBranch ("3")  
ProcessRunSqlQuery ("")
```

Parent topic: [Query setup actions](#)

Related reference:
[ProcessRunSqlQuery](#)

QuerySetDateFormat

Sets custom Date format for SQL queries.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetDateFormat (string dateFormat)
```

Parameters

dateFormat
Type: string

Parameters

dateFormat: Date format. Smart parameters are supported.

Returns

Always True.

Level

Any level.

Details

Configures the Date format used by the database. If this action is not called, the value for the Date format will be "yyyy/MM/dd". Detailed description of available values are:

- **d** Represents the day of the month as a number from 1 through 31. A single-digit day is formatted without a leading zero
- **dd** Represents the day of the month as a number from 01 through 31. A single-digit day is formatted with a leading zero
- **ddd** Represents the abbreviated name of the day of the week (Mon, Tues, Wed etc.)
- **dddd** Represents the full name of the day of the week (Monday, Tuesday etc.)
- **h** 12-hour clock hour (e.g. 7)
- **hh** 12-hour clock, with a leading 0 (e.g. 07)
- **H** 24-hour clock hour (e.g. 19)
- **HH** 24-hour clock hour, with a leading 0 (e.g. 19)
- **m** Minutes
- **mm** Minutes with a leading zero
- **M** Month number
- **MM** Month number with leading zero
- **MMM** Abbreviated Month Name (e.g. Dec)
- **MMMM** Full month name (e.g. December)
- **s** Seconds
- **ss** Seconds with leading zero
- **t** Abbreviated AM / PM (e.g. A or P)
- **tt** AM / PM (e.g. AM or PM)
- **y** Year, no leading zero (e.g. 2001 would be 1)
- **yy** Year, leading zero (e.g. 2001 would be 01)
- **yyy** Year (e.g. 2001 would be 001)
- **yyyy** Year (e.g. 2001 would be 2001)
- **K** Represents the time zone information of a date and time value (e.g. +05:00)
- **z** With DateTime values, represents the signed offset of the local operating system's time zone from Coordinated Universal Time (UTC), measured in hours. (e.g. +6)
- **zz** As z but with leading zero (e.g. +06)
- **zzz** With DateTime values, represents the signed offset of the local operating system's time zone from UTC, measured in hours and minutes. (e.g. +06:00)
- **f** Represents the most significant digit of the seconds fraction; that is, it represents the tenths of a second in a date and time value.
- **ff** Represents the two most significant digits of the seconds fraction; that is, it represents the hundredths of a second in a date and time value.
- **fff** Represents the three most significant digits of the seconds fraction; that is, it represents the milliseconds in a date and time value.
- **ffff** Represents the four most significant digits of the seconds fraction; that is, it represents the ten thousandths of a second in a date and time value. While it is possible to display the ten thousandths of a

second component of a time value, that value may not be meaningful. The precision of date and time values depends on the resolution of the system clock. On Windows NT 3.5 and later, and Windows Vista operating systems, the clock's resolution is approximately 10-15 milliseconds.

- **ffff** Represents the five most significant digits of the seconds fraction; that is, it represents the hundred thousandths of a second in a date and time value. While it is possible to display the hundred thousandths of a second component of a time value, that value may not be meaningful. The precision of date and time values depends on the resolution of the system clock. On Windows NT 3.5 and later, and Windows Vista operating systems, the clock's resolution is approximately 10-15 milliseconds.
- **ffffff** Represents the six most significant digits of the seconds fraction; that is, it represents the millionths of a second in a date and time value. While it is possible to display the millionths of a second component of a time value, that value may not be meaningful. The precision of date and time values depends on the resolution of the system clock. On Windows NT 3.5 and later, and Windows Vista operating systems, the clock's resolution is approximately 10-15 milliseconds.
- **fffffff** Represents the seven most significant digits of the seconds fraction; that is, it represents the ten millionths of a second in a date and time value. While it is possible to display the ten millionths of a second component of a time value, that value may not be meaningful. The precision of date and time values depends on the resolution of the system clock. On Windows NT 3.5 and later, and Windows Vista operating systems, the clock's resolution is approximately 10-15 milliseconds.
- **F** Represents the most significant digit of the seconds fraction; that is, it represents the tenths of a second in a date and time value. Nothing is displayed if the digit is zero.
- **:** Represents the time separator defined in the current `DateTimeFormatInfo...:TimeSeparator` property. This separator is used to differentiate hours, minutes, and seconds.
- **/** Represents the date separator defined in the current `DateTimeFormatInfo...:DateSeparator` property. This separator is used to differentiate years, months, and days.
- **"** Represents a quoted string (quotation mark). Displays the literal value of any string between two quotation marks ("). Your application should precede each quotation mark with an escape character (\).
- **'** Represents a quoted string (apostrophe). Displays the literal value of any string between two apostrophe (') characters.
- **%c** Represents the result associated with a c custom format specifier, when the custom date and time format string consists solely of that custom format specifier. That is, to use the d, f, F, h, m, s, t, y, z, H, or M custom format specifier by itself, the application should specify %d, %f, %F, %h, %m, %s, %t, %y, %z, %H, or %M. For more information about using a single format specifier, see [Using Single Custom Format Specifiers](#).

Example:

```
QuerySetDateFormat ("dd-MMM-yy")
```

Parent topic: [Query setup actions](#)

Related reference:

[ProcessRunSqlQuery](#)

[QuerySetDateTimeFormat](#)

QuerySetDateRange

Sets Date range for SQL query.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetDateRange (string start, string end, bool queryAgeStart)
```

Parameters

start
Type: string

end
Type: string

queryAgeStart
Type: bool

Parameters

- start: Date Range Start. Smart parameters are supported.
- end: Date Range End. Smart parameters are supported.
- queryAgeStart: True will use the batch start date. False will use the batch end date.

Returns

True if the query has been successfully set. It does not mean the query has been performed. Otherwise, False.

Level

Any level.

Details

Selects all of the batches that match the specified priority. If queryAgeStart is set to True, the query uses the batch start date, qs_start. If queryAgeStart is set to False, the query uses the batch end date, qu_done.

An exclamation point can be prefixed to the value to find all values except the ones specified.

Example:

```
QuerySetDateRange ("03/16/2010", "05/26/2010", "True")  
ProcessRunSqlQuery ("")
```

This example selects all of the batches that were created between the two specified dates.

```
QuerySetDateRange ("03/16/2010", "@DATE (mm/dd/yyyy) ", "True")  
ProcessRunSqlQuery ("")
```

This example uses a Smart parameter to obtain the current date. The query selects all batches that have been created between 03/16/2010 and the current date.

Parent topic: [Query setup actions](#)

Related reference:
[ProcessRunSqlQuery](#)

QuerySetDateTimeFormat

Sets custom DateTime format for SQL queries.

Member of namespace

Syntax

```
bool QuerySetDateTimeFormat (string dateTimeFormat)
```

Parameters

dateTimeFormat
Type: string

Parameters

dateTimeFormat: DateTime format. Smart parameters are supported.

Returns

Always True.

Level

Any level.

Details

Configures the Date and Time format used by the database. If this action is not called, the value for the Date format is yyyy/MM/dd HH:mm:ss. Detailed description of available values are:

- **d** Represents the day of the month as a number from 1 through 31. A single-digit day is formatted without a leading zero
- **dd** Represents the day of the month as a number from 01 through 31. A single-digit day is formatted with a leading zero
- **ddd** Represents the abbreviated name of the day of the week (Mon, Tues, Wed etc.)
- **dddd** Represents the full name of the day of the week (Monday, Tuesday etc.)
- **h** 12-hour clock hour (e.g. 7)
- **hh** 12-hour clock, with a leading 0 (e.g. 07)
- **H** 24-hour clock hour (e.g. 19)
- **HH** 24-hour clock hour, with a leading 0 (e.g. 19)
- **m** Minutes
- **mm** Minutes with a leading zero
- **M** Month number
- **MM** Month number with leading zero
- **MMM** Abbreviated Month Name (e.g. Dec)
- **MMMM** Full month name (e.g. December)
- **s** Seconds
- **ss** Seconds with leading zero
- **t** Abbreviated AM / PM (e.g. A or P)
- **tt** AM / PM (e.g. AM or PM)
- **y** Year, no leading zero (e.g. 2001 would be 1)
- **yy** Year, leading zero (e.g. 2001 would be 01)
- **yyy** Year (e.g. 2001 would be 001)
- **yyyy** Year (e.g. 2001 would be 2001)
- **K** Represents the time zone information of a date and time value (e.g. +05:00)

- **z** With DateTime values, represents the signed offset of the local operating system's time zone from Coordinated Universal Time (UTC), measured in hours. (e.g. +6)
- **zz** As z but with leading zero (e.g. +06)
- **zzz** With DateTime values, represents the signed offset of the local operating system's time zone from UTC, measured in hours and minutes. (e.g. +06:00)
- **f** Represents the most significant digit of the seconds fraction; that is, it represents the tenths of a second in a date and time value.
- **ff** Represents the two most significant digits of the seconds fraction; that is, it represents the hundredths of a second in a date and time value.
- **fff** Represents the three most significant digits of the seconds fraction; that is, it represents the milliseconds in a date and time value.
- **ffff** Represents the four most significant digits of the seconds fraction; that is, it represents the ten thousandths of a second in a date and time value. While it is possible to display the ten thousandths of a second component of a time value, that value may not be meaningful. The precision of date and time values depends on the resolution of the system clock. On Windows NT 3.5 and later, and Windows Vista operating systems, the clock's resolution is approximately 10-15 milliseconds.
- **fffff** Represents the five most significant digits of the seconds fraction; that is, it represents the hundred thousandths of a second in a date and time value. While it is possible to display the hundred thousandths of a second component of a time value, that value may not be meaningful. The precision of date and time values depends on the resolution of the system clock. On Windows NT 3.5 and later, and Windows Vista operating systems, the clock's resolution is approximately 10-15 milliseconds.
- **ffffff** Represents the six most significant digits of the seconds fraction; that is, it represents the millionths of a second in a date and time value. While it is possible to display the millionths of a second component of a time value, that value may not be meaningful. The precision of date and time values depends on the resolution of the system clock. On Windows NT 3.5 and later, and Windows Vista operating systems, the clock's resolution is approximately 10-15 milliseconds.
- **fffffff** Represents the seven most significant digits of the seconds fraction; that is, it represents the ten millionths of a second in a date and time value. While it is possible to display the ten millionths of a second component of a time value, that value may not be meaningful. The precision of date and time values depends on the resolution of the system clock. On Windows NT 3.5 and later, and Windows Vista operating systems, the clock's resolution is approximately 10-15 milliseconds.
- **F** Represents the most significant digit of the seconds fraction; that is, it represents the tenths of a second in a date and time value. Nothing is displayed if the digit is zero.
- **:** Represents the time separator defined in the current DateTimeFormatInfo / TimeSeparator property. This separator is used to differentiate hours, minutes, and seconds.
- **/** Represents the date separator defined in the current DateTimeFormatInfo / DateSeparator property. This separator is used to differentiate years, months, and days.
- **"** Represents a quoted string (quotation mark). Displays the literal value of any string between two quotation marks ("). Your application should precede each quotation mark with an escape character (\).
- **'** Represents a quoted string (apostrophe). Displays the literal value of any string between two apostrophe (') characters.
- **%c** Represents the result associated with a c custom format specifier, when the custom date and time format string consists solely of that custom format specifier. That is, to use the d, f, F, h, m, s, t, y, z, H, or M custom format specifier by itself, the application should specify %d, %f, %F, %h, %m, %s, %t, %y, %z, %H, or %M. For more information about using a single format specifier, see Using Single Custom Format Specifiers.

Example:

```
QuerySetDateFormat("dd-MMM-yy hh:mm:ss tt")
```

Parent topic: [Query setup actions](#)

Related reference:

[ProcessRunSqlCommand](#)

QuerySetGeneric

Builds an SQL query using the provided column name and value.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetGeneric (string column, string value)
```

Parameters

column

Type: string

value

Type: string

Parameters

- column: Column in the database table to query. Smart parameters are supported.
- value: Value to match within the specified column. Smart parameters are supported.

Returns

True if the query has been successfully set. It does not mean the query has been performed. Otherwise, False.

Level

Any level.

Details

Allows building of a query where the column and value of the tmbatch, queue or qstats table are explicitly specified.

Any wild card specified in the value parameter would need to be appropriate for your target database. The column value is not validated until the query is run using ProcessRunSqlQuery.

Example:

```
QuerySetGeneric("pb_userid", "admin")  
ProcessRunSqlQuery("")
```

Parent topic: [Query setup actions](#)

Related reference:

[ProcessRunSqlQuery](#)

QuerySetJobID

Sets the Job ID for the SQL query.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetJobID (string jobid)
```

Parameters

jobid
Type: string

Parameters

jobid: Job name. Smart parameters are supported.

Returns

True, if the query is successfully set. It does not mean that the query has been performed. Otherwise, True.

Level

Any level.

Details

When you are building a query, this action sets the job ID that is selected in the result set. If you want to match multiple job IDs, use commas to separate values. An exclamation point can be used to negate the query.

Example:

```
QuerySetJobID("Demo Job, Web Job")  
ProcessRunSqlQuery("")
```

This example creates a result set that contains batches with job ID of `Demo Job` and `Web Job`.

```
QuerySetJobID("Demo Job")  
QuerySetJobID("Web Job")  
ProcessRunSqlQuery("")
```

This example creates an empty result set.

```
QuerySetJobID("!Demo Job")  
ProcessRunSqlQuery("")
```

This example creates a result set that contains all batches except for the job ID of `Demo Job`.

Parent topic: [Query setup actions](#)

Related reference:
[ProcessRunSqlQuery](#)

QuerySetOperator

Sets the operator for the SQL query.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetOperator (string operator)
```

Parameters

operator
Type: string

Parameters

operator: The operator ID. Smart parameters are supported.

Returns

True if the query has been successfully set. It does not mean the query has been performed. Otherwise, False.

Level

Any level.

Details

Selects all of the batches that match the specified operator ID.

To find multiple values, separate them with commas. An exclamation point can be prefixed to the value to find all values except the ones specified.

Example:

```
QuerySetOperator ("Admin")  
ProcessRunSqlQuery ("")
```

Parent topic: [Query setup actions](#)

Related reference:
[ProcessRunSqlQuery](#)

QuerySetPriority

Sets the Priority for the SQL query.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetPriority (string priority)
```

Parameters

priority
Type: string

Parameters

priority: Batch Priority.

Returns

True if the query has been successfully set. It does not mean the query has been performed. Otherwise, False.

Level

Any level.

Details

Selects all of the batches that match the specified priority.

To find multiple values, separate them with commas. An exclamation point can be prefixed to the value to find all values except the ones specified.

Example:

```
QuerySetPriority("1")  
ProcessRunSqlQuery("")
```

Parent topic: [Query setup actions](#)

Related reference:
[ProcessRunSqlQuery](#)

QuerySetSeparator

Sets SQL Date and Time Separator for SQL queries.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetSeparator (string separator)
```

Parameters

separator
Type: string

Parameters

separator: Date separator. Smart parameters are supported.

Returns

Always True.

Level

Any level.

Details

Configures the Date and Time separator that is used by the database. If this action is not called, the value for the separator is selected based on the connection string. Some database separators are MS SQL " ' ", Oracle " ' ", Access "#".

Example:

```
QuerySetSeparator("#")
```

Parent topic: [Query setup actions](#)

Related reference:
[ProcessRunSqlQuery](#)

QuerySetStation

Sets the station for the SQL query.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetStation (string station)
```

Parameters

station
Type: string

Parameters

station: The station ID. Smart parameters are supported.

Returns

True if the query has been successfully set. It does not mean the query has been performed. Otherwise, False.

Level

Any level.

Details

Selects all of the batches that match the specified station.

To find multiple values, separate them with commas. An exclamation point can be prefixed to the value to find all values except the ones specified.

Example:

```
QuerySetStation("1")
ProcessRunSqlQuery("")
```

Parent topic: [Query setup actions](#)

Related reference:

[ProcessRunSqlQuery](#)

QuerySetStatus

Sets the Task status for the SQL query.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetStatus (string status)
```

Parameters

status

Type: string

Parameters

status: The batch status: aborted, cancelled, finished, hold, job done, pending, running. Smart parameters are supported.

Returns

True if the query has been successfully set. It does not mean the query has been performed. Otherwise, False.

Level

Any level.

Details

Selects all of the batches that match the specified status.

To find multiple values, separate them with commas. An exclamation point can be prefixed to the value to find all values except the ones specified.

Example:

```
QuerySetStatus("finished")
ProcessRunSqlQuery("")
```

Parent topic: [Query setup actions](#)

Related reference:

[ProcessRunSqlQuery](#)

QuerySetTaskID

Sets the Task ID for the SQL query.

Member of namespace

Maintenance Manager

Syntax

```
bool QuerySetTaskID (string taskid)
```

Parameters

taskid

Type: string

Parameters

taskid: Task ID. Smart parameters are supported.

Returns

True if the query has been successfully set. It does not mean the query has been performed. Otherwise, False.

Level

Any level.

Details

When building a query, this action sets the task ID that is selected in the result set. The current task batches selected will match the supplied value.

To find multiple values, separate them with commas. An exclamation point can be prefixed to the value to find all values except the ones specified.

Example:

```
QuerySetTaskID("Verify")
ProcessRunSqlQuery("")
```

This example queries all batches for the survey application that are at the verify task.

```
QuerySetTaskID("@B.MyQueryTask")
ProcessRunSqlQuery("")
```

This example uses Smart parameters to retrieve the name of the task to query from a batch level variable called *MyQueryTask*.

Parent topic: [Query setup actions](#)

Related reference:
[ProcessRunSqlQuery](#)

Batch processing actions

Use the Batch processing actions to run the SQL query and complete actions on the selected database records and, optionally, the corresponding batches.

The ProcessRunSqlQuery action runs the current query string and generates a recordset that contains information about all matching batches. For example, if you build a query by using `QuerySetStatus("hold")` and your query locates one batch with status hold. The returned recordset contains the following information that is aggregated from the tmbatch, qstats, and queue tables.

```
<rs:data xmlns:rs="urn:schemas-microsoft-com:rowset"><z:row pb_adjustdocs="0"
pb_adjustpages="0" pb_batch="20100260.001"
pb_batchdir="C:\Datacap\APT\batches\20100260.001" pb_expectdocs="0" pb_expectpgs="8"
pb_headertable=" " pb_ndocs="0" pb_needMeet="0" pb_pagefile="rrsvscan.xml"
pb_pages="8" qs_elaps="8" qs_op="admin"
qs_gid="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcaca00
4_3" qs_start="2010-09-17T07:35:26"
qs_station="2" qs_stop="2010-09-17T07:35:34"
qs_taskid="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcac
a004_VScan" qs_tsorder="0"
qu_admDB="156" qu_batch="20100260.001" qu_counter="0" qu_done="2010-09-17T07:35:34"
qu_elaps="8"
qu_id="_dcs_markdown_workspace_Transform_htmlout_0_com.ibm.dc.reference.doc_dcaca004
_3" qu_job="Demo" qu_lock="none" qu_parent="0" qu_priority="5"
qu_spawntype="0" qu_start="2010-09-17T07:35:26" qu_status="hold" qu_task="VScan"
qu_tsorder="0" xmlns:z="#RowsetSchema" /></rs:data>
```

The other actions in the batch processing category can manipulate the batches that are identified in the query results record set.

- [ProcessChangeBatchStatus](#)
Changes the status of one or more batches.
- [ProcessChangeBatchStatusOrder](#)
Changes batch status and order.
- [ProcessChangeBatchStatusTaskOrder](#)
Changes batch status, task and order.
- [ProcessClearAuditTable](#)
Clears the Audit table located in the admin database.
- [ProcessClearDebugTable](#)
Clears the Debug table located in the engine database.
- [ProcessDeleteBatchesEx](#)
Delete selected batches.
- [ProcessInjectBatches](#)
Injects the data from master batch to all batches selected and updates DB.
- [ProcessMoveBatchesEx](#)
Move selected batches to folder specified in the parameter.
- [ProcessMoveDBRecords](#)
Creates connection to the application based on the parameters that are provided and moves selected database data to this application.

- [ProcessResetPendingOrNotify](#)
Resets all selected batches to `Pending` status.
- [ProcessRunSqlQueryEx](#)
Runs the previously defined Maintenance Manager query. Use this action as an alternative to `ProcessRunSqlQuery`.

Parent topic: [Maintenance Manager actions](#)

ProcessChangeBatchStatus

Changes the status of one or more batches.

Member of namespace

Maintenance Manager

Syntax

```
bool ProcessChangeBatchStatus (string newStatus)
```

Parameters

`newStatus`
Type: string

Parameters

`newStatus`: The new batch status: aborted, cancelled, finished, hold, job done, pending, running. Smart parameters are supported.

Returns

True if the batch status is successfully changed. Otherwise, False.

Level

Batch level.

Details

Using the results from a previous query performed by Maintenance Manager actions, the selected batches have their status attribute changed.

The database connection is closed by this action.

Example:

```
QuerySetStation("1")  
QuerySetJobID("!Demo Job")  
ProcessRunSqlQuery("")  
ProcessChangeBatchStatus("hold")
```

Parent topic: [Batch processing actions](#)

ProcessChangeBatchStatusOrder

Changes batch status and order.

Member of namespace

Maintenance Manager

Syntax

```
bool ProcessChangeBatchStatusOrder (string newStatus, int newOrder)
```

Parameters

newStatus
Type: string
newOrder
Type: int

Parameters

- newStatus: The new batch status: aborted, cancelled, finished, hold, job done, pending, running. Smart parameters are supported.
- newOrder: New task order, zero-based index of task inside its job.

Returns

True if the batch status is successfully changed. Otherwise, False.

Level

Batch level.

Details

Using the results from a previous query performed by Maintenance Manager actions, the selected batches have their status and order attributes changed.

The database connection is closed by this action.

Example:

```
ProcessChangeBatchStatusOrder("hold", "1")
```

Parent topic: [Batch processing actions](#)

ProcessChangeBatchStatusTaskOrder

Changes batch status, task and order.

Member of namespace

Syntax

```
bool ProcessChangeBatchStatusTaskOrder (string newStatus, int newOrder, string newTask)
```

Parameters

newStatus
Type: string

newOrder
Type: int

newTask
Type: string

Parameters

- newStatus: The new batch status: aborted, cancelled, finished, hold, job done, pending, running. Smart parameters are supported.
- newOrder: The order of the new task, zero-based index of task inside its job.
- newTask: The name of the new task.

Returns

True if the batch status is successfully changed. Otherwise, False.

Level

Batch level.

Details

Using the results from a previous query performed by Maintenance Manager actions, the selected batches have their status, order and task attributes changed.

The database connection will be closed by this action.

Example:

```
QuerySetStation("1")  
QuerySetJobID("!Demo Job")  
ProcessRunSqlQuery("")  
ProcessChangeBatchStatusTaskOrder("hold", "1", "Verify")
```

Parent topic: [Batch processing actions](#)

ProcessClearAuditTable

Clears the Audit table located in the admin database.

Member of namespace

Maintenance Manager

Syntax

```
bool ProcessClearAuditTable ()
```

Parameters

None.

Returns

True if the table is cleared. Otherwise, False.

Level

Any level.

Details

Clears the Audit table located in the admin database. This action can be called to ensure that Audit table is cleared when information is moved from one database to another.

Example:

```
ProcessClearAuditTable ("")
```

Parent topic: [Batch processing actions](#)

Related reference:

[ProcessClearDebugTable](#)

ProcessClearDebugTable

Clears the Debug table located in the engine database.

Member of namespace

Maintenance Manager

Syntax

```
bool ProcessClearDebugTable ()
```

Parameters

None.

Returns

True if the table is cleared. Otherwise, False.

Level

Any level.

Details

Clears the Debug table located in the engine database. This action can be called to ensure that Debug table is cleared when information is moved from one database to another.

Example:

```
ProcessClearDebugTable ("")
```

Parent topic: [Batch processing actions](#)

Related reference:

[ProcessClearAuditTable](#)

ProcessDeleteBatchesEx

Delete selected batches.

Member of namespace

Maintenance Manager

Syntax

```
bool ProcessDeleteBatchesEx (bool deleteSubFolders, bool preserveEngineDBRecords)
```

Parameters

deleteSubFolders

Type: bool

preserveEngineDBRecords

Type: bool

Parameters

- deleteSubFolders: True deletes any sub-folders within batch directories regardless of creation source.
- preserveEngineDBRecords: True retains engine database records (only batch directories are deleted).

Returns

True if the batches are deleted. Otherwise False.

Level

Any level.

Details

Using the results from a previous query performed by Maintenance Manager actions, the selected batches are deleted from disk.

A previous query must have been run to identify the batches to delete. The database connection is closed by this action.

Example:

```
QuerySetStation("1")
QuerySetJobID("!Demo Job")
ProcessRunSqlQuery("")
ProcessDeleteBatchesEx("false, false")
```

Parent topic: [Batch processing actions](#)

Related reference:

[ProcessMoveBatchesEx](#)

[ProcessMoveDBRecords](#)

ProcessInjectBatches

Injects the data from master batch to all batches selected and updates DB.

Member of namespace

Maintenance Manager

Syntax

```
bool ProcessInjectBatches (string masterBatchID)
```

Parameters

masterBatchID
Type: string

Parameters

masterBatchID: Master batch to copy data from.

Returns

True if all the all the batches were successfully set. Otherwise, False.

Level

Any level.

Details

Allows creating large set of batches at some predefined state.

Example:

```
QuerySetBatchRange("20110059.001", "20110059.010")
InjectBatches("20110059.001")
```

Parent topic: [Batch processing actions](#)

Related reference:

[QuerySetBatchRange](#)

ProcessMoveBatchesEx

Move selected batches to folder specified in the parameter.

Member of namespace

Maintenance Manager

Syntax

```
bool ProcessMoveBatchesEx (string pathTo, bool moveSubFolders, bool preserveEngineDBPaths, bool continueOnError)
```

Parameters

pathTo
Type: string

moveSubFolders
Type: bool

preserveEngineDBPaths
Type: bool

continueOnError
Type: bool

Parameters

- pathTo: The destination directory for the batches. Smart parameters are supported.
- moveSubFolders: True moves any subfolders within batch directories regardless of creation source. This parameter is required if the batches contain subfolders or else operations, including database updates, will fail.
- preserveEngineDBPaths: True preserves original batch directory paths inside the engine database.
- continueOnError: True continues processing if any single batch fails to be moved with non-fatal error.

Returns

True, if the batches are successfully moved. Otherwise, False.

Level

Any level.

Details

Using the results from a previous query that is performed by Maintenance Manager actions, the selected batches are moved to the specified destination directory. A previous query must first be run to identify the batches to move to the new directory. The destination directory must exist. The database connection is closed by this action.

If you are moving batches from one location to another and also moving database records from one database to another, move the batches first. The movement of the database records must be done last.

Examples

The following example runs a query to select all batches from station "1" where the Job ID does not equal "Demo Job", then moves the batches to a directory called "My Old Batches".

```
QuerySetStation("1")
QuerySetJobID("!Demo Job")
ProcessRunSqlQuery("")
ProcessMoveBatchesEx("@STRING(C:\My Old Batches\), false, false, false")
```

This following example selects the same set of batches but it uses a Smart parameter to obtain the full path to directory from the Application Service. A key called BatchArchiveDirectory must exist in the Application Service. If these rules are used for multiple applications, each application can have a unique definition of this value within the Application Service, making these rules flexible.

One benefit of this approach is that in an environment where there is a test system and a production system, each system will have unique values stored in the Application Service. Each of the environments can have a different physical directory specified within the Application Service, allowing the rules in each system to remain identical.

```
QuerySetStation("1")
QuerySetJobID("!Demo Job")
ProcessRunSqlQuery("")
ProcessMoveBatchesEx("@APPPATH(BatchArchiveDirectory), false, false, false")
```

Parent topic: [Batch processing actions](#)

Related reference:

[ProcessDeleteBatchesEx](#)

[ProcessMoveDBRecords](#)

ProcessMoveDBRecords

Creates connection to the application based on the parameters that are provided and moves selected database data to this application.

Member of namespace

Maintenance Manager

Syntax

```
bool ProcessMoveDBRecords (string application, string server, string admin, string engine, bool debugFlag, string user, string password, string station, bool deleteOriginal, string targetDBSQLSeparator)
```

Parameters

application
Type: string

server
Type: string

admin
Type: string

engine
Type: string

debugFlag
Type: bool

user
Type: string
password
Type: string
station
Type: string
deleteOriginal
Type: bool
targetDBSQLSeparator
Type: string

Parameters

- application: Application name (Optional). Default value: the name of the application that is running this action.
- server: Server name (Optional). Default value: first available server in the application.
- admin: Administration database name (Optional). Default value: first available Admin DB.
- engine: Engine database name (Optional). Default value: first available Engine DB.
- debugFlag: Debug Flag (Optional). Default value: false.
- user: User name (Optional). Default value: current user credentials.
- password: Password (Optional). Default value: current user credentials.
- station: Station ID (Optional). Default value: Current station name
- deleteOriginal: Delete database records in source database.
- targetDBSQLSeparator: Target database date and time separator (Optional). If blank, that action tries to detect the correct separator that is based on the connection string.

Returns

Always True.

Level

Any level.

Details

Creates connection to the application based on the parameters that are provided and moves selected database data to this application. When you are performing operations on a batch and you are moving the database records from one database to another, do the database move operation last. When database records are moved, the database connection is still connected to the original database. Subsequent operations are on the original database, not the records in the new database.

Example:

```
ProcessMoveDBRecords("APTBack","tms","admin","engine","false","admin","admin","1","true","")
```

In this example, the source database is Oracle and the target is Access.

Parent topic: [Batch processing actions](#)

Related reference:

[ProcessMoveBatchesEx](#)

[ProcessDeleteBatchesEx](#)

ProcessResetPendingOrNotify

Resets all selected batches to `Pending` status.

Member of namespace

Maintenance Manager

Syntax

```
bool ProcessResetPendingOrNotify (int threshold, bool asattachment, string addressFrom, string addressTo, string subject, string user, string password, string domain, string server, string port)
```

Parameters

`threshold`
Type: int

`asattachment`
Type: bool

`addressFrom`
Type: string

`addressTo`
Type: string

`subject`
Type: string

`user`
Type: string

`password`
Type: string

`domain`
Type: string

`server`
Type: string

`port`
Type: string

Parameters

All parameters, except for `threshold`, support Smart parameters.

- `threshold`: Maximum number of attempts to reset the batch to status 'Pending'.
- `asattachment`: Send the log as attachment to the email.
- `addressFrom`: Address From (Optional). Default value: `currentUser@currentDomain`.
- `addressTo`: The email recipient. If multiple recipients, separate by using a comma.
- `subject`: Email subject (Optional). Default value: `'NENU notification current_date_and_time'`.
- `user`: Mail user name (Optional). Default value: current user credentials.
- `password`: Mail user password (Optional). Default value: current user credentials.
- `domain`: Mail user domain (Optional). Default value: current domain.
- `server`: Mail Server name (Optional). Default value: `mail.current_domain`.
- `port`: Mail Server Port number (Optional). Default value: 25.

Returns

True, if the email is sent. Otherwise, False.

Level

Any level.

Details

Resets the status of all batches that are selected by a previous SQL query to Pending status. The action attempts to reset the batch by the amount that is specified in the threshold parameter. If the maximum reset attempts are reached, the action sends an email, along with the log, to the provided list of recipients.

Example:

```
QuerySetStation("1")
QuerySetJobID("!Demo Job")
ProcessRunSqlQuery("")
ProcessResetPendingOrNotify("3", "rrodrig@somewhere.com",
"tom@somewhere.com,john@somewhere.com","","","","","","")
```

Parent topic: [Batch processing actions](#)

ProcessRunSqlQueryEx

Runs the previously defined Maintenance Manager query. Use this action as an alternative to ProcessRunSqlQuery.

Member of namespace

Maintenance Manager

Syntax

```
bool ProcessRunSqlQueryEx (int minRecords, int maxRecords)
```

Parameters

minRecords

The minimum number of record results that are required for this action to return True.

maxRecords

The maximum number of record results that are required for this action to return True. To indicate no maximum, specify -1 for the parameter value.

Returns

True in the following circumstances:

- The query is successful.
- The number of record results is within the specified range.

Otherwise, this action returns False.

Level

Any level.

Details

Use this action as an alternative to `ProcessRunSqlQuery` for finding batches. Unlike `ProcessRunSqlQuery`, this action can return `False` if the number of found batches is not as expected. The behavior of this action is the same as `ProcessRunSqlQuery` in the following ways:

Query definition	Use the query setup actions to define the SQL query that this action runs. For example, use <code>QueryClear</code> to clear the SQL query. To output the query statement, use <code>LogWriteSQLQuery</code> .
Result set processing	Use other batch processing actions to act upon the records that are returned by this action. To write out the result set for debugging purposes, use <code>LogWriteRecordSet</code> .

Example:

```
QuerySetStation("1")
QuerySetJobID("!Demo Job")
ProcessRunSqlQueryEx(1,-1)
```

In this example, the query results consist of the batches that were run on station “1” except for any “Demo Job” batches. If at least one such batch is found, the `ProcessRunSqlQueryEx` action returns `True`.

Parent topic: [Batch processing actions](#)

Related reference:

[Batch processing actions](#)

[LogWriteRecordSet](#)

[LogWriteSQLQuery](#)

[Query setup actions](#)

Logging actions

Use the Logging actions to write information to the Maintenance Manager and Windows log files and to send emails that contain the internal log file.

During rule execution, Maintenance Manager writes status messages to an internal log file and the Rulerunner log file.

- The internal log file is maintained in memory and is used by the `SendEmail` action.
- The Rulerunner log file is stored in the `application_name > batches > Maintenance Manager` folder.
- [LogClear](#)
Clears current Log.
- [LogConfigure](#)
Configures features of aTM logging.
- [LogSendEmail](#)
Sends email with log to comma-separated list of recipients.
- [LogWriteEventLog](#)
Writes a message to the Event Log.
- [LogWriteRecordSet](#)
Outputs the results of `ProcessRunSqlQuery` to the error log.

- [LogWriteSQLQuery](#)
Outputs the constructed SQL query to the error log.

Parent topic: [Maintenance Manager actions](#)

LogClear

Clears current Log.

Member of namespace

Maintenance Manager

Syntax

```
bool LogClear ()
```

Parameters

None.

Returns

Always True.

Level

Any level.

Details

Clears current log.

Example:

```
LogClear ()
```

Parent topic: [Logging actions](#)

LogConfigure

Configures features of aTM logging.

Member of namespace

Maintenance Manager

Syntax

```
bool LogConfigure (int severity, string filePath, bool overwrite, bool reflash, bool showTime, bool showDate, bool showSeverity)
```

Parameters

severity

Type: int

The log severity limit [0-9]. 0 = maximum logging. 4 or 5 = typically informational or error logging.

filePath

Type: string

Log file path name or blank to disable logging. Smart parameters are supported.

overwrite

Type: bool

True overwrites any existing file.

reflash

Type: bool

True flushes the error message buffer to disk after every write.

showTime

Type: bool

True adds the current time to each log message.

showDate

Type: bool

True adds the current date to each log message.

showSeverity

Type: bool

True adds the severity to the log.

Returns

True if logging is successfully configured. Otherwise, False.

Level

Any level.

Details

Configures aTM logging. The severity controls verbosity of the log; logging only severe errors will provide the best performance at expense of reduced debugging information, should a debug trace be required.

This will not change logging as configured for each of the tasks in the target application (i.e. RRS logs). If this action is not called, no aTM log file will be created. aTM logging can be evaluated to see which requests were transmitted to the Maintenance Manager Server.

Example:

```
LogConfigure("5", "@STRING(C:\ParentDir\NENU\Logs\NENU.aTM)")
```

In this example, rolling 'NENU.aTM.#.log' logs is placed into the custom directory.

```
LogConfigure("5", "@APPPATH(export)")
```

In this example, the output directory is retrieved from the application service using Smart parameters. The path is the export directory for the current application.

Parent topic: [Logging actions](#)

LogSendEmail

Sends email with log to comma-separated list of recipients.

Member of namespace

Maintenance Manager

Syntax

```
bool LogSendEmail (bool asattachment, string addressFrom, string addressTo, string subject, string user, string password, string domain, string server, string port)
```

Parameters

asattachment
 Type: bool
addressFrom
 Type: string
addressTo
 Type: string
subject
 Type: string
user
 Type: string
password
 Type: string
domain
 Type: string
server
 Type: string
port
 Type: string

Parameters

All parameters support Smart parameters.

- asattachment: Send the log as attachment to the email.
- addressFrom: Address From (Optional). Default value: currentUser@currentDomain.
- addressTo: Comma-separated list of recipients.
- subject: email subject (Optional). Default value: 'NENU notification current_date_and_time'.
- user: Mail user name (Optional). Default value: current user credentials.
- password: Mail user password (Optional). Default value: current user credentials.
- domain: Mail user domain (Optional). Default value: current domain.
- server: Mail Server name (Optional). Default value: mail.current_domain.
- port: Mail Server Port number (Optional). Default value: 25.

Returns

True, if the email is sent. Otherwise, False.

Level

Any level.

Details

Sends an email by using the SMTP protocol. As Maintenance Manager actions run, an in-memory log tracks each of the Maintenance Manager actions that are called and their parameters. The LogSendEmail action places the action activity information into an email and send it.

Example:

```
LogSendEmail("jsmith@somewhere.com", "jdoe@somewhere.com,mmoore@somewhere.com",  
            "", "", "", "", "", "", "")
```

Parent topic: [Logging actions](#)

LogWriteEventLog

Writes a message to the Event Log.

Member of namespace

Maintenance Manager

Syntax

```
bool LogWriteEventLog (string message, int level, int eventID)
```

Parameters

message
 Type: string
level
 Type: int
eventID
 Type: int

Parameters

- message: Message to write to the Event Log and the local log. Smart parameters are supported.
- level: Integer value. 0 - informational, 1 - Warning, 2 - Error.
- eventID: Integer value. Desired Event ID.

Returns

True if the event is successfully logged. Otherwise, False.

Level

Any level.

Details

Unconditionally writes out a message to the Windows Event Log and to the application log file for the running task. Set the level and event ID to the values that is appropriate for the message being logged.

Example:

```
LogWriteEventLog("This is an informational message.", "0", "10")
```

This example writes the message `This is an informational message.` to the event log.

```
LogWriteEventLog("@B.MyMessage", "0", "10")
```

This example uses Smart parameters to write out the value of the batch level variable `MyMessage`.

Parent topic: [Logging actions](#)

LogWriteRecordSet

Outputs the results of `ProcessRunSqlQuery` to the error log.

Member of namespace

Maintenance Manager

Syntax

```
bool LogWriteRecordSet ()
```

Parameters

None.

Returns

True if the write is successful. Otherwise, False.

Level

Any level.

Details

Writes out the result of the last call to `ProcessRunSqlQuery` to the error file. This action can be useful for debugging an application.

Example:

```
QuerySetStation("1")  
QuerySetJobID("!Demo Job")  
ProcessRunSqlQuery("")  
LogWriteRecordSet("")
```

Parent topic: [Logging actions](#)

Related reference:

[ProcessRunSqlQuery](#)

[LogWriteSQLQuery](#)

LogWriteSQLQuery

Outputs the constructed SQL query to the error log.

Member of namespace

Maintenance Manager

Syntax

```
bool LogWriteSQLQuery ()
```

Parameters

None.

Returns

True if the log is successfully written. Otherwise, False.

Level

Any level

Details

Writes out the result of the previous calls to the "QuerySet" actions to the error file. This action can be useful for debugging an application, allowing you to view the exact SQL that was constructed and used in ProcessRunSqlQuery.

Example:

```
QuerySetStation("1")  
QuerySetJobID("!Demo Job")  
LogWriteSQLQuery("")  
ProcessRunSqlQuery("")
```

Parent topic: [Logging actions](#)

Related reference:

[ProcessRunSqlQuery](#)

[LogWriteRecordSet](#)

Reporting actions

Use the Reporting actions to write information to the report tables in the Engine database for use by Datacap Report Viewer.

The Reporting actions can query the active users on an application and set the database tables that contain the reports for processed batches and users.

- [ReportQueryTMUsage](#)
Update the ReportUser Database with the current users.
- [ReportSetReportingTable](#)
Sets database that contains reports on all processed batches
- [ReportSetUsageDBTable](#)
Sets database that contains reports on users that are logged in to Datacap.

Parent topic: [Maintenance Manager actions](#)

ReportQueryTMUsage

Update the ReportUser Database with the current users.

Member of namespace

Maintenance Manager

Syntax

```
bool ReportQueryTMUsage ()
```

Parameters

None.

Returns

True if the database update was successful. Otherwise, False.

Level

Any level.

Details

This action will query the number of users current logged on and place the information into the reportUser. This information is statistical information that can later be used to generate usage reports with the Report Viewer reporting system.

Example:

```
ReportQueryTMUsage ("")
```

Parent topic: [Reporting actions](#)

ReportSetReportingTable

Sets database that contains reports on all processed batches

Member of namespace

Maintenance Manager

Syntax

```
bool ReportSetReportingTable (string tbName, string batchColumn, string attemptColumn, string doneColumn, string actionColumn)
```

Parameters

tbName

Type: string

Table name in Engine database.

batchColumn

Type: string

Optional. Name of the column that contains Batch ID.

attemptColumn

Type: string

Optional. Name of the column that contains attempts.

doneColumn

Type: string

Optional. Name of the column that contains completion result.

actionColumn

Type: string

Optional. Name of the column that contains last action that is performed.

Returns

Always True.

Level

Batch level.

Details

Sets database that contains reports on all processed batches. Table must exist in the engine database.

Example:

```
ReportSetReportingTable("NENU", "nn_batch", "nn_attempt", "nn_done", "nn_action")
```

Parent topic: [Reporting actions](#)

Related reference:

[ReportSetUsageDBTable](#)

ReportSetUsageDBTable

Sets database that contains reports on users that are logged in to Datacap.

Member of namespace

Maintenance Manager

Syntax

```
bool ReportSetUsageDBTable (string tbName, string ipAddressColumn, string jobIDColumn, string portColumn, string processedBathcesColumn, string stationColumn, string taskIDColumn, string userIDColumn, string queryTimeColumn)
```

Parameters

tbName

Type: string
Table name in Engine database.

ipAddressColumn
Type: string
IP address column.

jobIDColumn
Type: string
Job ID column.

portColumn
Type: string
Port column.

processedBatchesColumn
Type: string
Processed batches column.

stationColumn
Type: string
Station ID column.

taskIDColumn
Type: string
Task ID column.

userIDColumn
Type: string
User ID column.

queryTimeColumn
Type: string
Query time column.

Returns

Always True.

Level

Batch level.

Details

Sets database that contains reports on users that are logged in to Datacap. Table must exist in the engine database.

Example:

```
ReportSetUsageDBTable("reportUsers","ru_ip","ru_job","ru_port","ru_bathces","ru_station","ru_task","ru_user","ru_time")
```

Parent topic: [Reporting actions](#)

Related reference:

[ReportSetReportingTable](#)

OCR_J actions

The OCR_J actions use the Rosetta Stone Japanese OCR engine to perform OCR for Japanese machine print and Japanese handwriting.

- [InitializeEngine](#)
Initializes the Datacap interface for the Rosetta Stone Japanese OCR engine. Call this action before you call other actions in the OcrRose library.
- [Recognize](#)
Performs field-zone OCR for Japanese machine print and Japanese handwriting.
- [ReleaseEngine](#)
Releases the Datacap interface for the Rosetta Stone Japanese OCR engine so that all interface resources are freed. Do not call other actions in the OCR_J library after you call this action.

Parent topic: [Global actions](#)

InitializeEngine

Initializes the Datacap interface for the Rosetta Stone Japanese OCR engine. Call this action before you call other actions in the OcrRose library.

Member of namespace

OCR_J

Syntax

```
bool InitializeEngine ()
```

Returns

True if the action succeeds. Otherwise, the action returns False.

Level

All levels.

Details

Example:

```
InitializeEngine ()
```

Parent topic: [OCR_J actions](#)

Recognize

Performs field-zone OCR for Japanese machine print and Japanese handwriting.

Member of namespace

OCR_J

Syntax

```
bool Recognize ()
```

Returns

True if the action succeeds. Otherwise, the action returns False.

Level

Field level.

Details

Use this action to perform field-zone OCR for Japanese machine print and Japanese handwriting. As indicated by the following tables, you can configure this action's OCR behavior by setting the specified DCO variables before you call this action.

Example:

```
InitializeEngine ()
rrSet ("3", "@F.rsFrameType")
rrSet ("8", "@F.rsOcrType")
Recognize ()
ReleaseEngine ()
```

Frame type

The DCO variable `rsFrameType` specifies the frame type of the fields for which OCR is to be performed. The frame type is the way that fields are framed on the form by boxes, ticks, vertical lines, or other marks. Here are the valid values:

0	None (unconstrained field)
1	Normal
2	Individual
3	Field
4	Box
5	Ladder
6	Solid ladder

The default frame type is "0".

OCR type

The DCO variable `rsOcrType` specifies the type of the characters for which OCR is to be performed. Here are the valid values:

1	Machine printed Japanese alpha characters
2	Machine printed numbers
4	Machine printed alpha-numeric characters
8	Handwritten numbers
16	Handwritten alpha-numeric characters
32	Handwritten Japanese alpha characters

The default OCR type is “1”.

Parent topic: [OCR_J actions](#)

ReleaseEngine

Releases the Datacap interface for the Rosetta Stone Japanese OCR engine so that all interface resources are freed. Do not call other actions in the OCR_J library after you call this action.

Member of namespace

OCR_J

Syntax

```
bool ReleaseEngine ()
```

Returns

True if the action succeeds. Otherwise, the action returns False.

Level

All levels.

Details

Example:

```
ReleaseEngine ()
```

Parent topic: [OCR_J actions](#)

OCR_A actions

Use the OCR_A actions to do text recognition by using the FineReader OCR engine.

- [EnableEngineLogsOCR_A](#)
Enables OCR_A Engine logging
- [OCRA_ConvertImage2BW](#)
This action converts a Color or Grayscale image to Black and White.
- [Recognize](#)
Recognize refers to settings in the OCR/A tab of the Recognition Options Setup dialog to recognize all characters on a page, and populates the page's Fingerprint file (.cco) with the recognition results.
- [RecognizeBarcodeOCR_A](#)
A field/page level action that retrieves barcodes.
- [RecognizeFieldOCR_A](#)
A field-level action that retrieves a zoned field's settings from the OCR/A tab of the Recognition Options Setup dialog, and uses these settings to recognize the field's value.
- [RecognizeFieldVoteOCR_A](#)
A field-level action that initiates a voting procedure that first uses specifications in the OCR/A tab of the Recognition Options Setup dialog to recognize the field's characters.

- [RecognizePageFieldsOCR_A](#)
A page-level action that recognizes all fields on the page that have been configured for OCR/A recognition (see the OCR/A tab of the Recognition Options Setup dialog.)
- [RecognizePageOCR_A](#)
Refers to settings in the OCR/A tab of the Recognition Options Setup dialog to recognize all characters on a page, and populates the page's Fingerprint file (.cco) with the recognition results.
- [RecognizeToALTOOCR_A](#)
Converts a scanned Images (.tif) to an ALTO electronic Document Format (XML) file.
- [RecognizeToPDFOCR_A](#)
Converts a scanned Images (.tif) to an Adobe Portable Document Format (PDF) file.
- [ReleaseEngineOCR_A](#)
This action releases the OCR_A engine.
- [RotateImageOCR_A](#)
Use with `Recog_Shared > RotateTIO` to update the CCO file with the correct position coordinates after image rotation.
- [SetAutoRotationOCR_A](#)
This action set to False turns off automatic image orientation detection and rotation.
- [SetConfCalculationParamsOCR_A](#)
Specifies the values to use for OCR_A->Datacap confidence mapping.
- [SetFastModeOCR_A](#)
This action set to TRUE provides 2-2.5 times faster recognition speed at the cost of a moderately increased error rate (1.5-2 times more errors).

Parent topic: [Global actions](#)

EnableEngineLogsOCR_A

Enables OCR_A Engine logging

Member of namespace

ocr_a

Syntax

```
bool EnableEngineLogsOCR_A ()
```

Parameters

None.

Returns

True, if logging is successfully enabled. Otherwise, False.

Level

Any level.

Details

This action enables OCR_A Engine logging. The log file is created in the batch folder with the name engine.log.

Example:

```
RecognizePageOCR_A()  
EnableEngineLogsOCR_A()
```

This sequence creates a log file in the batch folder with the name engine.log.

Parent topic: [OCR_A actions](#)

OCRA_ConvertImage2BW

This action converts a Color or Grayscale image to Black and White.

Member of namespace

ocr_a

Syntax

```
bool OCRA_ConvertImage2BW (StrParam)
```

Parameters

The file extension that the action is to assign to the backup of the original Image file. For example: tio.

The extension should be 3 or 4 alphanumeric characters.

Returns

False if called at a level other than the Page. False if the parameter is not 3 or 4 alphanumeric characters. Otherwise, True.

Level

Page.

Details

This action converts a Color or Grayscale image to Black and White.

Example:

```
OCRA_ConvertImage2BW()
```

Parent topic: [OCR_A actions](#)



Recognize

Recognize refers to settings in the OCR/A tab of the Recognition Options Setup dialog to recognize all characters on a page, and populates the page's Fingerprint file (.cco) with the recognition results.

The *Recognize* action performs document layout analysis and OCR, generating a layout XML file such as TM000001_layout.xml. The layout file groups text into blocks as a person would be looking at the document.

Each block might have the default type of block or a specific type such as title or table. There are *locate* actions available in the *DocumentAnalytics* action library to navigate the block structure such as "GoSiblingBlockNext". This is in contrast to the CCO file produced by other actions that groups text into lines that span the width of the page. The layout XML file also retains font and color attributes, which are saved in CSS format, for the text that is used for extracting data and reconstructing the document in a new format. To use the Locate actions and perform click "n" key during verification, use the *CreateCcoFromLayout* action in the *SharedRecognitionTools* action library. This action creates a CCO file for the page after producing the layout XML file.

Member of namespace

ocr_a

Syntax

```
bool Recognize ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a **Page** object of the Document Hierarchy. Otherwise, **True**.

Level

Page level.

Details

The following are the types of elements that might be present in the layout XML file:

Table 1. Block types versus XML node

Block type	Node in the layout XML file
Block	Block
Header	Header
Footer	Footer
Title	Title
Heading1	H1
Heading2	H2
Heading3	H3
Picture	Picture
Barcode	Barcode
Space	S
Tab	Tab
Table	Table

Block type	Node in the layout XML file
Row	Row
Cell	Cell
Paragraph	Para
Line	L
Sentence	Sent
Word	W
Character	C

Supported File Formats

The Recognize action can process color images and PDF files. When processing PDF documents, the action extracts embedded text within the PDF document, and performs recognition only on those areas that contain data but do not contain embedded text. This improves the processing speed and overall performance of the processing of PDF documents.

Tip: It is also possible to process Microsoft Excel, Microsoft Word, Html, Rtf, and Txt documents by first converting those documents to searchable PDFs via the Convert library. For example, a Microsoft Excel document can be converted to PDF by calling the action *ExcelWorkbookToPdf*. After the PDF document is created, it can be processed through the Recognize action.

Language detection

Language detection helps improve recognition results. Instead of using the default English setting, it detects the language and results in more accurate OCR results. When the OCR process is complete, a report on the number of languages detected (and total number of words that are detected for each language) is generated. This report is stored in the runtime DCO as variables, and can also be found in the layout XML file. To enable automatic language detection:

1. Use *rrSet* or a similar action to set the *y_lg* variable to a list of comma-separated list of **at least** three supported auto-detection languages.
2. After specifying the list of languages, call the *Recognize* action.

Important: The list of languages must be minimized to the languages expected to be processed by the application. More languages slow down the processing. However, if an application process only two languages, you must still provide at least three languages to enable automatic language detection.

Languages Supported by Auto Detection

Important: When you set the comma-separated list of languages, make sure that the languages are entered exactly as written in this list. An invalid language name causes the action to abort.

Note: Text that follows the colon ":" is informational only and must not be included.

- Arabic: Arabic (Saudi Arabia)
- ArmenianWestern: Armenian (Western)
- AzeriLatin: Azerbaijani (Latin)
- Bashkir
- Bulgarian
- Catalan
- ChinesePRC: Chinese Simplified
- ChinesePRC+English: Chinese Simplified and English
- ChineseTaiwan: Chinese Traditional
- ChineseTaiwan+English: Chinese Traditional and English

- Croatian
- Czech
- Danish
- Dutch
- English
- Estonian
- Finnish
- French
- German
- GermanNewSpelling: German (new spelling)
- GermanLuxembourg: German (Luxembourg)
- Greek
- Hebrew
- Hungarian
- Hungarian
- Indonesian
- Italian
- Japanese
- Korean
- Korean+English: Korean and English
- KoreanHangul: Korean (Hangul)
- Latin
- Latvian
- Lithuanian
- Mixed: Russian and English
- Norwegian: NorwegianNynorsk and NorwegianBokmal
- NorwegianBokmal: Norwegian (Bokmal)
- NorwegianNynorsk: Norwegian (Nynorsk)
- OldEnglish
- OldFrench
- OldGerman
- OldItalian
- OldSpanish
- Polish
- PortugueseBrazilian
- PortugueseStandard
- Romanian
- RussianOldSpelling
- Russian
- RussianWithAccent
- Slovak
- Slovenian
- Spanish
- Swedish
- Tatar
- Thai
- Turkish
- Ukrainian
- Vietnamese

The language can be bound to the DCO object by selecting it in the OCR_A tab in the Zones tab of Datacap Studio. When selected in the OCR_A tab, the variable `y_lg` is set to the language. The language also can be set within rules by using the `rrSet` action to set the `y_lg` variable to the wanted language.

For example: `rrSet("Italian", "@X.y_lg")` Sets the language to Italian.

If the `s_lg` variable is not set for the current DCO object, the recognized language is determined by the current locale set with the `hr_locale` variable. For example, if the locale is set for Germany, `rrSet("de-DE", "@X.hr_locale")`, then the text is recognized as German.

If both the `hr_locale` and `y_lg` variables are set, the value in `y_lg` takes precedence over the locale setting. If `y_lg` is set but the engine should use the value set for `hr_locale` instead, setting the variable `dco_uselocale` to "1" gives precedence to `hr_locale`.

If the page you are recognizing is formatted for right-to-left text, such as Arabic or Hebrew, then the variable "`hr_bidi`" must be set to "`rtl`" to indicate that the page is right to left.

Example:

```
rrSet("English,French,German", @P.y_lg) - Auto detection of English, French or German
rrSet("English,German,GermanNewSpelling,Norwegian", @P.y_lg) - Auto detection of
English, German, or Norwegian
rrSet("ChinesePRC+@CHR(43)+English", "@P.y_lg") - An exception for specification of
Simplified Chinese and English
```

Custom Parameters

The following variables can be used to set custom parameters for recognition:

`y_userProfile`

Reserved for Internal use only.

`y_predefinedProfile`

Set this variable to the name of a predefined profile to use during recognition. Valid values are:

- *DocumentConversion_Accuracy*
- *DocumentConversion_Speed*
- *DocumentArchiving_Accuracy*
- *DocumentArchiving_Speed*
- *BookArchiving_Accuracy*
- *BookArchiving_Speed*
- *TextExtraction_Accuracy*
- *TextExtraction_Speed*
- *EngineeringDrawingsProcessing*
- *BusinessCardsProcessing*

Text Extraction versus Text Recognition

When a PDF is recognized, by default, the text included in the recognition results is obtained from a combination of automatic recognition that is run on the PDF and from searchable text that is embedded within the PDF.

Any images that are embedded on the page have the text that is recognized by the engine. If areas of the page contain both an image and searchable text that is associated with the image, the engine decides whether the engine must use the searchable text or recognize the text from the matching image.

Because the engine performs recognition, the confidence of the text might vary even if the same searchable text is embedded in the PDF.

The variable `y_contentReuseMode` can be used to force the engine only to use the recognized text on the page or only to use the embedded text on the page. One reason why you might decide only to use the embedded text is to prevent recognition and produce high confidence results.

A drawback of only using the embedded text is that if the embedded text is wrong or incomplete, recognition is not performed to capture that missing data that results in a layout XML that is incomplete compared to what the user sees when the user views the PDF. Do not use this setting if the source PDF file is of the image-on-text type because in this case, the text layer is not extracted.

If a text line contains characters that are not included in the alphabet of the selected recognition languages, this text is not written to the result and mode 0 or 1 must be used.

These settings of `y_contentReuseMode` can be set on the DCO node that is being converted:

`rrSet("0", "@X.y_contentReuseMode")` - The default auto mode that uses a combination of recognition and embedded text.

`rrSet("1", "@X.y_contentReuseMode")` - Only recognition is used to create the layout XML.

`rrSet("2", "@X.y_contentReuseMode")` - Only embedded text is used to create the layout XML.

Example:

This sequence creates a layout XML file, and subsequently, a CCO file for the current page. Auto detection is enabled for English, French, and Japanese documents. The CCO file that is produced is ready for use by navigation and pattern match actions.

```
rrSet("English,French,Japanese", "@P.y_lg")
Recognize()
CreateCcoFromLayout()
```

Table Identification

Some recognition engines can identify a table on a document when performing full page recognition when using layout files produced by the *Recognize* action. When text is recognized as a table, it means that additional metadata is internally stored about the words that have been recognized. This extra metadata stores the cell information, row and column position, for the text. This table metadata can be used by subsequent actions that support table functions. This is an alternative approach to using the "line items" feature that processes recognized data as tables without a table being identified by the recognition engine.

The *Recognize* action attempts to detect tables on a page, identifying the rows and columns. If a table is detected, then the table structure can be used by subsequent actions that manipulate the recognized tables. As with character recognition, 100% accuracy of a table's rows and columns is not guaranteed. If this feature is needed, it is advisable to test on various tables that you expect to be processing in your application, and determine whether the accuracy is good enough to provide the functions that you need. If table recognition accuracy is not good enough for your wanted approach, it is recommended that you change your approach.

It is possible that parts of a table might not be recognized as expected even if they visually look good to the human eye. For example, parts of your table might be left outside of the table structure or rows and columns might be combined in ways that might not be expected from visually looking at the table. If the tables in your documents are not being identified well by the engine, then you might consider a different approach, such as using line items, to process tabular data within your documents.

The following tips can help to improve the recognition of your table:

- Tables must be well-defined with grid lines that identify the rows and columns of the table.
- Do not perform line removal if you are attempting to detect table structures.
- Cells cannot intersect each other.
- All cells must have a rectangular shape.

Forcing single lines per table cell

If your table contains only a single row but the engine recognizes multiple rows per table cell, the engine can be instructed to recognize the multiple rows as a single row by setting the DCO variable *y_SingleLinePerCell* to "1".

Example: `rrSet("1", "@X.y_SingleLinePerCell")`

Splitting table by separators

While identifying a table layout, the engine uses grid lines and its own heuristics to determine the rows and columns for the table. Sometimes this can cause the cells created by the engine to be different from what is defined by grid lines on the page. For example, your table may have grid lines that show multiple lines in a cell but instead those lines are interpreted as single line cells. The DCO variable *y_SplitOnlyBySeparators* can be set to "1" which tells the engine to use grid lines only while identifying the layout of a table. For example: `rrSet("1", "@X.y_SplitOnlyBySeparators")`. When enabled, the setting tells the engine to use the grid lines to guide the identification, providing a table layout that closely matches the visible grid lines. This setting is off by default.

If *y_SplitOnlyBySeparators* is enabled, the engine does not attempt to recognize a table without grid lines.

Disabling table detection

Table detection is enabled by default. You can disable it by setting the page level variable *y_DetectTables* to "0".

Example: `rrSet("0", "@X.y_DetectTables")`.

However, typically, it is not necessary to disable table detection. However, the setting is provided in the case it gives better results for your document types.

Zoning the table location

In some cases, the table identification can be improved by identifying the table location / zone. This might improve table identification when the engine is not detecting the table boundaries well or when the table does not have gridlines to identify the table boundaries.

If the page can be fingerprinted so that the location of the table is always known, then the table area can be zoned to identify the table boundaries. To identify the table zone to the recognition engine, specify the name of the field that contains the table zone by using the variable *y_TableZone*. For example, if the current page has a field that is called "MyTable" and the zone for that field identifies the table location on the page then the action `rrSet("MyTable", "@X.y_TableZone")` allows the engine to use the zone to identify the table.

If a table is recognized by using the user-provided zone, sub-fields are created off of the *y_TableZone* specified field with the table contents. If there are multiple tables on the page, only one table can be identified by using this method.

If you need to identify multiple tables on a page, then you must use auto detection, process the page multiple times with different zones for each table or take a different approach. If the location of the table cannot be predicted or identified before recognition, then you would need to use auto detection or take a different approach.

Providing the engine with the table zone gives better results in situations where table auto detect is making mistakes, as in the case when there are no grid lines. Be aware that this technique still cannot guarantee 100% accuracy and you might need to take additional custom steps to massage the recognized data for your needs. It is a similar situation as to the need to correct standard character recognition.

Parent topic: [OCR_A actions](#)

Related reference:

[Recognize](#)

RecognizeBarcodeOCR_A

A field/page level action that retrieves barcodes.

Member of namespace

ocr_a

Syntax

```
bool RecognizeBarcodeOCR_A ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Field object or Page object of the Document Hierarchy. Otherwise, True.

Level

Field and Page levels.

Details

A field/page level action that retrieves barcodes.

Example:

```
TaxpayerSSN Rule 1  
RecognizeBarcodeOCR_A
```

Parent topic: [OCR_A actions](#)

RecognizeFieldOCR_A

A field-level action that retrieves a zoned field's settings from the OCR/A tab of the Recognition Options Setup dialog, and uses these settings to recognize the field's value.

Member of namespace

ocr_a

Syntax

```
bool RecognizeFieldOCR_A ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Field object of the Document Hierarchy. Otherwise, True.

Level

Field level.

Details

This field-level action retrieves a zoned field's settings from the OCR/A tab of the Recognition Options Setup dialog, and uses these settings to recognize the field's value.

Example:

```
TaxpayerSSN Rule 1  
RecognizeFieldOCR_A()
```

In the example, the rule uses the action to retrieve and apply settings in the OCR/A tab of the Recognition Options Setup dialog, settings that were previously assigned to a Document Hierarchy's zoned field.

Parent topic: [OCR_A actions](#)

RecognizeFieldVoteOCR_A

A field-level action that initiates a voting procedure that first uses specifications in the OCR/A tab of the Recognition Options Setup dialog to recognize the field's characters.

Member of namespace

ocr_a

Syntax

```
bool RecognizeFieldVoteOCR_A ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Field object of the Document Hierarchy. Otherwise, True.

Level

Field level.

Details

This field-level action initiates a voting procedure that first uses specifications in the OCR/A tab of the Recognition Options Setup dialog to recognize the field's characters.

When this action stores the results of recognition, it first determines if the corresponding Field object of the Document Hierarchy contains a value. If a value is present, the action compares the field's existing value with the recognition results - character by character.

If a particular character's values match, the Confidence Rating for the character is raised to the maximum level. If the values do not match, the Confidence Rating for the character is lowered to the minimum.

Note: When using this voting procedure, the second Recognition engine is secondary and its results are never assigned. Instead, the action changes the Confidence Ratings on the basis of results provided by the first Recognition engine. If there are no recognition results previous to this action, it will act just like the RecognizeFieldOCR_A action.

Example:

```
RecognizeFieldICR_C()  
RecognizeFieldVoteOCR_A()
```

Parent topic: [OCR_A actions](#)

RecognizePageFieldsOCR_A

A page-level action that recognizes all fields on the page that have been configured for OCR/A recognition (see the OCR/A tab of the Recognition Options Setup dialog.)

Member of namespace

ocr_a

Syntax

```
bool RecognizePageFieldsOCR_A ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Page object of the Document Hierarchy. Otherwise, True.

Level

Page level.

Details

OCR/A This page-level action recognizes all fields on the page that have been configured for OCR/A recognition (see the OCR/A tab of the Recognition Options Setup dialog.)

Note: Individual field-level recognition actions will overwrite the results from this page-level action. The action will not recognize a zoned field if the Skip Recognition checkbox in the OCR/A tab of the Recognition Options Setup dialog has been selected.

Example:

```
ReadZones ()  
RecognizePageFieldsOCR_A ()
```

Parent topic: [OCR_A actions](#)

RecognizePageOCR_A

Refers to settings in the OCR/A tab of the Recognition Options Setup dialog to recognize all characters on a page, and populates the page's Fingerprint file (.cco) with the recognition results.

Member of namespace

ocr_a

Syntax

```
bool RecognizePageOCR_A ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Page object of the Document Hierarchy. Otherwise, True.

Level

Page level.

Details

This action responds to settings in the OCR/A tab of the Recognition Options Setup dialog to recognize all characters on a page, and populates the page's CCO file with the recognition results. If a CCO file does not exist at the time this action is called, the action will create one.

Example:

```
AnalyzeImage ()  
RotateImage ()  
RecognizePageOCR_A ()
```

This sequence creates a CCO file for the current page, and checks to see if rotation of the image is needed. Full-page recognition then takes place in response to settings in the OCR/A tab of the Recognition Options Setup dialog. The recognition results are stored in the CCO file.

Parent topic: [OCR_A actions](#)

RecognizeToALTOOCR_A

Converts a scanned Images (.tif) to an ALTO electronic Document Format (XML) file.

Member of namespace

ocr_a

Syntax

```
bool RecognizeToALTOOCR_A()
```

Parameters

None.

Returns

False If called at an invalid level. Otherwise, True.

Level

Document or Page Level.

Details

Converts a scanned Images (.tif) to an ALTO Document Format (XML) file.

When placed at Page level, the action recognizes and converts the current tif page to an ALTO file.

When placed at Document level, the action recognizes and converts all tif pages in the existing doc into one ALTO file.

Document Format

The following variables can be used to set the properties of the ALTO xml file.

y_AltoFontFormattingMode

Specifies the character attributes that are saved to the ALTO xml file.

Valid values are.

- 0 - The only saved attribute is whether characters are subscript or superscript. This value is the default.
- 1 - The following attributes are saved, whether characters are subscript, superscript, bold, italic, underlined, strikethrough. Font size and font name are not saved.
- 2 - All font attributes are saved.

y_AltoWriteNondeskedCoordinates

Specifies whether character, word, block coordinates that are written into files in ALTO format are defined on an original image. Or are defined on an image that is used for recognition to which different modifications, such as deskewing, were applied. This property is True by default, which means that the coordinates are defined on the original page.

If you set this property to False, export to ALTO is run faster because there is no need to convert the coordinates between the modified image and the original image, which takes a long time. If this property is set to the default True value, the baseline position is not written during export. If it is set to False, the baseline position is written into the resulting ALTO file because ALTO format requires the baseline position be defined by

only one number. In the original coordinates, the baseline might not be strictly horizontal or vertical. In this case, it is impossible to define its position by a single number.

Document Contents

To exclude specific page types, set the variable *typesToExclude* to a comma delimited list of page types to exclude from the ALTO xml file.

To include specific page types, set the variable *typesToInclude* to a comma delimited list of page types to include in the ALTO xml file.

To exclude specific page status, set the variable *statusToExclude* to a comma delimited list of page status to exclude from the ALTO xml file.

When more than one filter is specified, the following order of precedence takes place:

- *statusToExclude* overrides *typesToInclude*
- *typesToInclude* overrides *typesToExclude*

If you are calling the action at the Document level, the types and status filters apply to both the documents and their child pages.

If you are calling the action at the Page level, the types and status filters apply to the page only.

The variables in the previous three sections must be set before you call the RecognizeToALTOOCR_A action.

Example

```
rrset ("75", "@D.statusToExclude")
rrSet ("Blank", "@D.typesToExclude")
RecognizeToALTOOCR_A()
```

This example creates an ALTO XML document with all of the pages that are contained in the DCO Document object except those pages with type "Blank" and status "75".

Parent topic: [OCR_A actions](#)

RecognizeToPDFOCR_A

Converts a scanned Images (.tif) to an Adobe Portable Document Format (PDF) file.

Member of namespace

ocr_a

Syntax

```
bool RecognizeToPDFOCR_A()
```

Parameters

None.

Returns

False if called at an invalid level. Otherwise, True.

Level

Document or Page Level.

Details

Converts one or more scanned Images (.tif) to an Adobe Portable Document Format (PDF) file. The PDF is searchable as it also includes the text as read directly by the recognition engine.

When placed at Page level, the action recognizes and converts the current TIF page to a PDF file.

When placed at document level, the action recognizes and converts all TIF pages in the existing doc into one PDF file. If the pages are PDF files, the action builds a new PDF that combines the page level PDFs into a single PDF.

Document Format

To create PDF/A1A documents, set the *y_pdfA* variable to "1" before you call `RecognizeToPDFOCR_A`.

To create PDF/A1B documents, set the *y_pdfA* variable to "1" and the *y_pdfA1B* variable to "1" before you call `RecognizeToPDFOCR_A`.

To set the MRC (Mixed Raster Content) Mode for conversion to PDF/A, set the *y_pdfMRCMode* variable to one of the following values:

- 0 - Engine decides whether MRC is to be used. Default.
- 1 - MRC is always used.
- 2 - MRC is never used. MRC technology uses a lossy compression algorithm. Some unimportant information from the source image (background texture, garbage, and so on) can be lost. Disable MRC if even insignificant information from the source image cannot be lost. Using a parameter of 2 helps to address issues where the text in the PDF document is too dark.

Document Contents

To exclude specific page types, set the variable *typesToExclude* to a comma-delimited list of page types to exclude from the PDF.

To include specific page types, set the variable *typesToInclude* to a comma-delimited list of page types to include in the PDF.

To exclude specific page status, set the variable *statusToExclude* to a comma-delimited list of page status to exclude from the PDF.

When more than one filter is specified, the following order of precedence takes place:

- *statusToExclude* overrides *typesToInclude*.
- *typesToInclude* overrides *typesToExclude*.

If you are calling the action at the Document level, the types and status filters apply to both the documents and their child pages.

If you are calling the action at the Page level, the types and status filters apply to the page only.

By default, recognition is performed on images, creating searchable text in the PDF. To prevent recognition, creating an image only PDF, set the variable *y_PDFImageOnly* to 1 in the current DCO object.

If you are creating a searchable PDF, for more information, on the supported languages and how to configure them, refer the OCR_A action, [Recognize](#).

Document Attributes

The following variables can be used to set the corresponding PDF document attributes:

- *y_PDFKeys*
- *y_PDFAuthor*
- *y_PDFTitle*
- *y_PDFSubject*
- *y_PDFProducer*
- *y_pdfCreator*
- *y_PDFQuality*
- *y_pdfDelTmp*

Memory or Disk Processing

By default the conversion is performed in memory. If you are creating PDF with many pages, it is possible for the conversion to run out of memory. The disk can be used for processing by setting the DCO variable *y_maxPagesForInMemoryProcessing* to the maximum number of pages for in-memory processing. If the document contains more pages than this value, the disk is used instead of memory.

The variables in the previous three sections must be set before you call the `RecognizeToPDFOCR_A` action.

Including PDF Annotations

By default, when you convert PDF to PDF, annotations in the source PDF file are not included in the output PDF. "Free Text" annotations in source PDF can be included in the output PDF by setting the page DCO variable *y_IncludeAnnotation* to "1". Other types of PDF annotations are not supported, such as popup and ink annotations. This feature does not cause the text of a "Sticky note" to be displayed on the image and a sticky note icon might display on the final image regardless of this setting.

PDF Export Optimization

The variable *y_pdfExportScenario* can be used to set the scenario of export to PDF (PDF/A) format, which optimizes export for some parameters. This impacts the size and quality of the output PDF.

It takes the following values:

- 0 - Optimize the PDF (PDF/A) export in order to receive the best quality of the resulting file.(This is default)
- 1 - The PDF (PDF/A) export will be balanced between the quality of the resulting file, its size and the time of processing.
- 2 - Optimize the PDF (PDF/A) export in order to receive the minimum size of the resulting file.
- 3 - Optimize the PDF (PDF/A) export in order to receive the highest speed of processing.

Example

```
rrset("IBM", "@D.y_PDFProducer")
rrSet("75", "@D.statusToExclude)
rrSet("Blank", "@D.typesToExclude)
RecognizeToPDF_A()
```

This example creates a PDF document with all of the pages that are contained in the DCO Document object except those pages with type "Blank" and status "75".

Parent topic: [OCR_A actions](#)

ReleaseEngineOCR_A

This action releases the OCR_A engine.

Member of namespace

ocr_a

Syntax

```
bool ReleaseEngineOCR_A ()
```

Returns

Always True.

Level

All.

Details

This action releases the OCR_A engine.

Example:

```
ReleaseEngineOCR_A()
```

Parent topic: [OCR_A actions](#)

RotateImageOCR_A

Use with `Recog_Shared > RotateTIO` to update the CCO file with the correct position coordinates after image rotation.

Member of namespace

ocr_a

Syntax

```
bool RotateImageOCR_A ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Page object of the Document Hierarchy, or if the action cannot locate the Image file representing the current page. Otherwise, True.

Level

Page only.

Details

This action checks if the scanned Image file needs to be rotated by 90, 180, or 270 degrees to be in the upright position. If rotation is necessary, the action saves the Image file in the new, correct position.

Note: You can specify a target language (for example, "Hebrew") so that the engine can properly rotate the text according to that language.

Example:

```
RotateImageOCR_A ()  
AnalyzeImage ()
```

Parent topic: [OCR_A actions](#)

SetAutoRotationOCR_A

This action set to False turns off automatic image orientation detection and rotation.

Member of namespace

ocr_a

Syntax

```
bool SetAutoRotationOCR_A (StrParam)
```

Parameters

True: Forces image orientation detection and rotation. This is the default value.

False: Image orientation and rotation will not be performed.

Returns

Always True.

Level

All.

Details

This action set to True forces image orientation detection and rotation. If this action is not called, the value will default to True. If used, this action must be called prior to recognition and both actions must be called at the same level.

Example:

```
SetAutoRotationOCR_A ("True")  
RecognizePageOCR_A
```

Parent topic: [OCR_A actions](#)

SetConfCalculationParamsOCR_A

Specifies the values to use for OCR_A->Datacap confidence mapping.

Member of namespace

ocr_a

Syntax

```
bool SetConfCalculationParamsOCR_A (StrParam)
```

Parameters

The M and C values for the following formula:

$$\text{Datacap Confidence} = \text{MAX}(10, (M/100) * (\text{OCR_A Confidence} + C))$$

The default values for M is 10. The default value for C is 60.

Returns

False if both parameters are not passed or are not numeric. Otherwise, True.

Level

Any level.

Details

Specifies the values to use for OCR_A->Datacap confidence mapping.

Example:

```
SetConfCalculationParamsOCR_A(0.1,70)
```

Parent topic: [OCR_A actions](#)

SetFastModeOCR_A

This action set to TRUE provides 2-2.5 times faster recognition speed at the cost of a moderately increased error rate (1.5-2 times more errors).

Member of namespace

ocr_a

Syntax

```
bool SetFastModeOCR_A (StrParam)
```

Parameters

True Enables Fast Mode which sacrifices recognition quality over speed.

False: Disables Fast Mode causing the recognition to run slower, but provides more accurate results.

If no parameter is specified, the value defaults to False.

Returns

Always True.

Level

All.

Details

This action set to TRUE provides 2-2.5 times faster recognition speed at the cost of a moderately increased error rate (1.5-2 times more errors).

It is recommended to disable fast mode if you are performing field level recognition because you will sacrifice quality yet see negligible speed increase at the field level.

If you use this action, it must be called prior to recognition.

Example:

```
SetFastModeOCR_A("True")
```

Parent topic: [OCR_A actions](#)

OCR_N actions

Use the OCR_N actions to do recognition by using the NovoDynamics engine. The OCR_N actions can run recognition on a full page or on all of the field zones that are defined for the current page.

If you plan to use the NovoDynamics engine for Arabic recognition, you must separately license the engine directly from the vendor and then install the engine on the machine that is running the recognition rules. In addition, you must complete these steps:

1. Copy the contents of the NovoDynamics bin folder into the Datacap\DCShared\OCRN folder.
Tip: The default location of the bin folder is C:\Program Files (x86)\NovoDynamics\NovoVerus\bin.
2. Use a command line to register the Datacap connector DLL:

```
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\RegAsm.exe  
datacap.libraries.novodynamics.dll /codebase
```

You can ignore a warning that the DLL is not signed.

Arabic language recognition works optimally when the scanned image has at least a 300-DPI resolution.

- [RecognizePageFieldsOCR_N](#)
Does full page recognition and populates the fingerprint (CCO) file of the page with the results.
- [RecognizePageOCR_N](#)
Does recognition on all field zones that are defined for the current page and writes the results to the runtime page data file.

Parent topic: [Global actions](#)

RecognizePageFieldsOCR_N

Does full page recognition and populates the fingerprint (CCO) file of the page with the results.

Member of namespace

Datacap.Libraries.NovoDynamics

Syntax

```
bool RecognizePageFieldsOCR_N ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Page object of the Document Hierarchy. Otherwise, True.

Level

Page only.

Details

This page-level action recognizes all fields on the page that have been configured for OCR/N recognition (see the action library help text for available settings via runtime\setup variables).

Important: Page level recognition settings are used to recognized the fields\zones. Per zone recognition settings are not supported.

Example:

```
ReadZones ()  
RecognizePageFieldsOCR_N ()
```

Parent topic: [OCR_N actions](#)

RecognizePageOCR_N

Does recognition on all field zones that are defined for the current page and writes the results to the runtime page data file.

Member of namespace

Datacap.Libraries.NovoDynamics

Syntax

```
bool RecognizePageOCR_N ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Page object of the Document Hierarchy. Otherwise, True.

Level

Page only.

Details

This action performs full page recognition.

The NormalizeCCO action from the CCO2CCO action library should be called after RecognizePageOCR_N if the application will be using the navigation and pattern match actions to find recognized text on a page or perform pattern matching.

If a CCO file does not exist at the time this action is called, the action will create one.

Example:

```
RotateImage ()
RecognizePageOCR_N ()
NormalizeCCO ("")
```

This sequence creates a CCO file for the current page, and checks to see if rotation of the image is needed. Full-page recognition then takes place in response to settings (see the action library help text for available settings via runtime\setup variables).

The recognition results are stored in the CCO file. The words and lines in the CCO are then sorted for use by navigation and pattern match actions.

Parent topic: [OCR_N actions](#)

OCR_SR actions

Use the OCR_SR actions to do recognition and image rotation by using the updated Nuance OmniPage OCR engine. You can run recognition actions on field zones and pages. The results can be processed by actions in other libraries and ultimately displayed to the user for verification. The results can also be written out to files in several possible file formats.

OCR/SR actions perform automatic retries when a recognition operation takes longer than expected. For information about the logic of the automatic retry mechanism and how to override the default settings that govern that logic, see [SetupAutomaticRetry](#).

After a recognition operation completes, the variable RecogStatus is set to indicate the success or failure of recognition. For page-level recognition, here are the possible RecogStatus values and their meanings:

0	Success.
1	Success but no results. The page was empty.
2	Success. Additional processing such as RotateImage was performed.
Any other value	Failure.

- [Recognize](#)
Performs document layout analysis and OCR and also generates a layout XML file such as TM000001_layout.xml.
- [RecognizeFieldOCR_S](#)
Does recognition on the zone of the current field and writes the result to the runtime page file.
- [RecognizeFieldVoteOCR_S](#)
Does recognition on the zone of the current field and compares the result to the existing field value, character by character. Raises the confidence level when the characters match and lowers it when they do not match.
- [RecognizePageFieldsOCR_S](#)
Does recognition on all field zones that are defined for the current page and writes the results to the runtime page data file.
- [RecognizePageOCR_S](#)
Does full page recognition and populates the fingerprint (CCO) file of the page with the results.
- [RecognizeToFileOCR_S](#)
Does full page recognition and writes the recognition results to one of several available output file types, such as .doc, .rtf, .html.
- [RecognizeToPDFOCR_S](#)
Does full page recognition and saves the current page as a PDF file. You can also create the file in PDF/A format.
- [RotateImageExOCR_S](#)
Rotates an image.
- [RotateImageOCR_S](#)
Use with the RotateTIO action from the Recog_Shared library to update the CCO file with the correct position coordinates after image rotation.
- [SetContinueOnFailureOCR_S](#)
Sets a value indicating whether a recognition action will abort or continue after it has encountered an error.
- [SetEngineTimeoutOCR_S](#)
Specifies the number of seconds to wait before it is determined that an OCR/S recognition action is not running properly.
- [SetOutOfProcessLoggingOCR_S](#)
Writes logs for the out-of-process process for debugging.
- [SetOutOfProcessTimeoutOCR_S](#)
Sets the number of seconds to wait to determine that an out-of-process action has stalled.
- [UseOutOfProcessRecogOCR_S](#)
Sets whether recognition will run out-of-process or in process. Running out of process improves stability of the application.

Parent topic: [Global actions](#)

Recognize

Performs document layout analysis and OCR and also generates a layout XML file such as TM000001_layout.xml.

Member of namespace

OCR_SR

Syntax

```
bool Recognize ()
```

Returns

False if the ruleset with this action is not bound to a page object of the document hierarchy. Otherwise, this action returns True.

Level

Page level.

Details

Performs document layout analysis and OCR and also generates an XML layout file such as TM000001_layout.xml.

The layout file groups text into blocks similar to how a person would see and identify the structure in the document. (In contrast, other actions produce a CCO file that groups text into lines that span the width of the page.) For example, a page can have items such as tables, paragraphs, and lines, which are all a type of a block. For information about the block types, see [DocumentAnalytics actions](#). A block might be of the default type or of a specific type such as title or table. The type depends on how the recognition engine interprets the block. Locate actions, such as GoSiblingBlockNext, are available in the Locate action library to navigate the block structure.

The layout XML file also retains the font and color attributes for the extraction text in CSS format. This text is used for extracting data and reconstructing the document in a new format.

Example:

In the following example, the Recognize action first creates an XML layout file, and then the CreateCcoFromLayout action creates a CCO file from that layout file. Both files are created for the current page.

```
Recognize ()
CreateCcoFromLayout ()
```

Supported File Formats

This action can process color images and PDF files. It processes PDF documents in the following manner:

- Extracts embedded text within the PDF document
- Performs recognition only on those areas that contain data but do not contain embedded text

This behavior improves the processing speed and overall performance of PDF document processing.

Tip: You can also process the following types of documents by first converting the documents to searchable PDFs with Convert library actions:

- Microsoft Excel
- Microsoft Word
- HTML
- RTF
- Txt

For example, you can convert a Microsoft Excel document to PDF by calling the action ExcelWorkbookToPdf. Once the PDF document is created, it can be processed with the Recognize action.

Language Detection

The Recognize action does not support language detection. However, this feature is available in the Recognize action of the OCR_A action library.

Parent topic: [OCR_SR actions](#)

Related reference:

[CreateCcoFromLayout](#)

[Recognize](#)

[DocumentAnalytics actions](#)

RecognizeFieldOCR_S

Does recognition on the zone of the current field and writes the result to the runtime page file.

Member of namespace

OCR_SR

Syntax

```
bool RecognizeFieldOCR_S ()
```

Parameters

None

Returns

False if the ruleset with this action is not bound to a Field object of the Document Hierarchy. Otherwise, True.

Level

Field level.

Details

This field-level action is a shortcut to zonal recognition procedures that are carried out in response to settings in the OCR/S tab of Datacap Studio.

This action supports the automatic retry mechanism.

Example

```
RecognizeFieldOCR_S ()
```

Parent topic: [OCR_SR actions](#)

Related reference:

[SetupAutomaticRetry](#)

RecognizeFieldVoteOCR_S

Does recognition on the zone of the current field and compares the result to the existing field value, character by character. Raises the confidence level when the characters match and lowers it when they do not match.

Member of namespace

OCR_SR

Syntax

```
bool RecognizeFieldVoteOCR_S ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Field object of the Document Hierarchy. Otherwise, True.

Level

Field only.

Details

This field-level action initiates a voting procedure that first uses specifications in the OCR/S tab of the Recognition Options Setup dialog to recognize the characters of the field.

When this action stores the results of recognition, it first determines if the corresponding Field object of the Document Hierarchy contains a value. If a value is present, the action compares the field's existing value with the recognition results, character by character.

If a particular character's values match, the Confidence Rating for the character is raised to 9 if the original confidence is smaller than 9. Otherwise the confidence of matching characters is raised to the maximum level (10).

When using this voting procedure, the second Recognition engine is secondary and its results are never assigned. Instead, the action changes the Confidence Ratings on the basis of results provided by the first Recognition engine. If there are no recognition results previous to this action, it acts like the RecognizeFieldOCR_S action.

This action supports the automatic retry mechanism.

Example

```
RecognizeFieldICR_C()  
RecognizeFieldVoteOCR_S()
```

Parent topic: [OCR_SR actions](#)

Related reference:

[SetupAutomaticRetry](#)

RecognizePageFieldsOCR_S

Does recognition on all field zones that are defined for the current page and writes the results to the runtime page data file.

Member of namespace

OCR_SR

Syntax

```
bool RecognizePageFieldsOCR_S ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Page object of the Document Hierarchy. Otherwise, True.

Level

Page only.

Details

This page-level action recognizes all fields on the page that have been configured for OCR/S recognition, see the OCR/S tab of the Recognition Options Setup dialog.

Note: Individual field-level recognition actions will overwrite the results from this page-level action.

The action does not recognize a zoned field if the Skip Recognition checkbox in the OCR/S tab of the Recognition Options Setup dialog is selected.

Example

```
ReadZones ()
RecognizePageFieldsOCR_S ()
```

Parent topic: [OCR_SR actions](#)

RecognizePageOCR_S

Does full page recognition and populates the fingerprint (CCO) file of the page with the results.

Member of namespace

OCR_SR

Syntax

```
bool RecognizePageOCR_S ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Page object of the Document Hierarchy. Otherwise, True.

Level

Page only.

Details

This action responds to settings in the OCR/S tab of the Recognition Options Setup dialog to recognize all characters on a page, and populates the page's CCO file with the recognition results.

Attention: The NormalizeCCO action from the CCO2CCO action library should be called after RecognizePageOCR_S if the application is using the navigation and pattern match actions to find recognized text on a page or perform pattern matching. If a CCO file does not exist when this action is called, the action creates one.

This action supports the automatic retry mechanism.

The DCO variable *s_charReplace* can be used to indicate characters that need to be replaced with an alternate character during the recognition step. This procedure can be used when replacing a specific character in a post-processing step is not feasible. Using this technique provides a way to limit the characters with full-page recognition as can be done with field-level recognition. The replacement string must contain pairs of comma-separated decimal values that represent the characters to replace, such as:

"original,new,original,new,original,new" and so on. It is recommended that this feature is used only if absolutely necessary. It is usually better to adjust characters later in a follow-on step.

The *s_charReplace* variable is only used during full-page recognition. It is not used for field-level recognition.

Attention: If a CCO file does not exist at the time when this action is called, the action will create one. If a CCO file already exists, it will be replaced.

Example

```
AnalyzeImage ()  
RotateImage ()  
RecognizePageOCR_S ()  
NormalizeCCO ("")
```

This sequence creates a CCO file for the current page, and checks to see whether the rotation of the image is needed. Full-page recognition then takes place in response to settings in the OCR/S tab of the Recognition Options Setup dialog. The recognition results are stored in the CCO file. The words and lines in the CCO are then sorted for use by navigation and pattern match actions.

Example

This example shows how to use the character replacement string.

```
rrSet ("8212,45", "@X.s_charReplace")  
RecognizePageOCR_S ()
```

In this case, any dashes that are recognized as a Unicode dash ("8213" in decimal) are converted to a standard minus sign character ("45" in decimal).

If there are more characters to filter, you should continue with more pairs in the same string.

Parent topic: [OCR_SR actions](#)

Related reference:
[SetupAutomaticRetry](#)

RecognizeToFileOCR_S

Does full page recognition and writes the recognition results to one of several available output file types, such as .doc, .rtf, .html.

Member of namespace

OCR_SR

Syntax

```
bool RecognizeToFileOCR_S (int FileType)
```

Parameters

FileType
Type int

Parameters

fileType - The action requires a Numeric parameter from 1-22 to specify a combination of recognition targets and output formats.

Important: Image refers to the image of the bound Page object of the Document Hierarchy. Filename is the string portion of a file's name that precedes its extension.

The output for all of these parameters will produce a file name that is identical to the original file name and will have the extension specified for that parameter.

1. A PDF document with the original image in the foreground with the recognized text hidden in the background (but in the correct position). Perfect for archiving and indexing documents.
2. A general PDF document where the text in the original image is replaced by the corresponding text recognized by the engine.
3. A special type of PDF document, where the suspect words are covered by their images cut out from the original image.
4. A non-searchable PDF document.
5. Recognize an HTML image of the bound Page object of the Document Hierarchy. Output .html (HTML 140).
6. Recognize an image of the bound Page object of the Document Hierarchy in an Excel file. Output .xls (Excel 2000.)
7. Recognize any image of the bound Page object of the Document Hierarchy in a WordML file. Output .doc (Word ML).
8. Recognize any image of the bound Page object of the Document Hierarchy in an RTF2000 file. Output .rtf (RTF 2000).
9. Recognize the image of the bound Page object of the Document Hierarchy in a Text file with an .RTF6 extension. Output .rtf (Rich Text).
10. Recognize the image of the Page object of the Document Hierarchy in a Text file with an .RTF6 extension. Output .rtf (Rich Text).
11. Recognize the image of the Page object of the Document Hierarchy in a Text file with an .Text extension. Output .txt (Text).
12. Recognize the image of the Page object of the Document Hierarchy in a Text file with an Csv extension. Output .txt (CSV - Comma Separated Variable).
13. Recognize the image of the Page object of the Document Hierarchy in a Text file with a .FormattedTxt extension. Output .txt (Formatted Text).

14. Recognize the image of the Page object of the Document Hierarchy in a Text file with a .UText extension. Output .txt (Text).
15. Recognize the image of the Page object of the Document Hierarchy in a Text file with a .UCSV extension. Output .CSV (Comma Separated Variable).
16. Recognize the image of the Page object of the Document Hierarchy in a Text file with a .UFormattedText extension. Output .txt (Text).
17. Recognize the image of the Page object of the Document Hierarchy in a Text file with an .Audio extension. Output aud (Text).
18. Recognize the image of the Page object of the Document Hierarchy in a Text file with a .WordPad extension. Output .rtf (Rich Text for WordPad).
19. Recognize the image of the Page object of the Document Hierarchy in a Text file with an .XML extension. Output .xml (XML).

Returns

False if a ruleset with this action is bound to a Field object of the Document Hierarchy, or if the parameter is not numeric. Otherwise, True.

Level

Page or Document.

Details

Performs OCR recognition on the image of a source page, and stores the output of the OCR/S recognition engine in a file. The output file is in one of 22 alternative formats. Because the files are not actually processed in the format you specify, this action is useful primarily for debugging the engine, or if you need raw (unverified) OCR output in that format.

By default, PDF documents created by this action are compatible with PDF Version 1.6.

However, it is possible to change the default compatibility by setting the *s_pdfVersion* variable to one of the following values:

```
2 = PDF Version 1.5
3 = PDF Version 1.4
4 = PDF Version 1.3
5 = PDF Version 1.2
6 = PDF Version 1.1
7 = PDF Version 1.0
8 = PDF Version A
9 = PDF Version 1.6
10 = PDF Version 1.7
11 = PDF Version A2B
12 = PDF Version A2U
13 = PDF Version A1A
14 = PDF Version A2A
```

To exclude specific page types, set the variable *typesToExclude* to a comma delimited list of page types to exclude from the pdf.

To exclude specific page status, set the variable *statusToExclude* to a comma delimited list of page status to exclude from the pdf.

These variables must be set before calling the action `RecognizeToPDFOCR_S`.

This action supports the automatic retry mechanism.

Example

The following example creates a PDF document with all pages contained in the dco document object except for pages with type `Blank` and status of 75:

```
rrSet ("75", "@D.statusToExclude")
rrSet ("Blank", "@D.typesToExclude")
RecognizeToFileOCR_S (1)
```

Parent topic: [OCR_SR actions](#)

Related reference:

[SetupAutomaticRetry](#)

RecognizeToPDFOCR_S

Does full page recognition and saves the current page as a PDF file. You can also create the file in PDF/A format.

Member of namespace

OCR_SR

Syntax

```
bool RecognizeToPDFOCR_S (int OutputPDFType)
```

Parameters

OutputPDFType
Type int

Parameters

A number value that indicates the PDF output type

1. A PDF document with the original image in the foreground with the recognized text hidden in the background (but in the correct position). Perfect for archiving and indexing documents.
2. A general PDF document where the text in the original image is replaced by the corresponding text that is recognized by the engine.
3. A special type of PDF document, where the suspect words are covered by their images cut out from the original image.
4. A non-searchable PDF document.

Returns

False if the rule with this action is not applied to a document or page object or if the parameters are not in the valid range. Otherwise, True.

Level

Document and Page only.

Details

This action converts a scanned Image file (.tif) to an Adobe Portable Document Format (PDF) file.

By default, PDF documents created by this action are compatible with PDF Version 1.6.

However, it is possible to change the default compatibility by setting the *s_pdfVersion* variable to one of the following values:

2 = PDF Version 1.5
3 = PDF Version 1.4
4 = PDF Version 1.3
5 = PDF Version 1.2
6 = PDF Version 1.1
7 = PDF Version 1.0
8 = PDF Version A
9 = PDF Version 1.6
10 = PDF Version 1.7
11 = PDF Version A2B
12 = PDF Version A2U
13 = PDF Version A1A
14 = PDF Version A2A

To exclude specific page types, set the variable *typesToExclude* to a comma delimited list of page types to exclude from the PDF.

To include specific page types, set the variable *typesToInclude* to a comma delimited list of page types to include in the PDF.

To exclude specific page status, set the variable *statusToExclude* to a comma delimited list of page status to exclude from the PDF.

When more than one filter is specified, the following order of precedence takes place:

- *statusToExclude* overrides *typesToInclude*
- *typesToInclude* overrides *typesToExclude*

If you are calling the action at the Document level, the types and status filters apply to both the documents and their child pages.

If you are calling the action at the Page level, the types and status filters apply to the page only.

To create a PDF in PDF/A format, set the variable *s_pdfVersion* to one of the following supported PDF/A formats:

- IMF_PDFA1B = 8
- IMF_PDFA2B = 11
- IMF_PDFA2U = 12
- IMF_PDFA1A = 13
- IMF_PDFA2A = 14

Important:

If you want to apply any of these variables (*typesToExclude*, *typesToInclude*, *statusToExclude*, *s_pdfVersion*), you must set the variables that you want before you call the RecognizeToPDFOCR_A action.

This action supports the automatic retry mechanism.

Example 1

```
rrSet ("75", "@D.statusToExclude")  
rrSet ("Blank", "@D.typesToExclude")
```

RecognizeToPDF(3)

This example creates a PDF document with all of the pages that are contained in the DCO Document object except those pages with type "Blank" and status "75".

Example 2

```
rrSet("8", "@P.s_pdfVersion")  
RecognizeToPDFOCR_S(1)
```

This example creates a PDF document in PDF/A-1b format. You can find `RecognizeToPDFOCR_S(1)` in the following area in the action library: `ocr_sr > Datcap.Libraries.ScansoftR.Actions > RecognizeToPDFOCR_S`.

Parent topic: [OCR_SR actions](#)

Related reference:

[SetupAutomaticRetry](#)

RotateImageExOCR_S

Rotates an image.

Member of namespace

OCR_SR

Syntax

```
bool RotateImageExOCR_S (string ImagePath, string RotationMode)
```

Parameters

ImagePath

The image file to rotate. If left blank, the current page image file is used by default. Smart parameters are supported.

RotationMode

The type of rotation to apply to the image. Here are the valid values:

0	Auto
1	No rotation
2	90-degree rotation, clockwise
3	180-degree rotation, clockwise
4	90-degree rotation, counter-clockwise

If left blank, Auto is used by default.

Returns

False if any of the following conditions apply:

- The ruleset with this action is not bound to a page object of the document hierarchy

- The action cannot locate the image file that you specified in the first parameter
- The rotation mode that you specified is not within the valid range of values

Otherwise, this action returns True.

Level

Page level.

Details

Use this action to rotate an image. After the action runs and the image is rotated, the applied rotation mode is saved in the page level variable OCRSRotMod.

If you do not specify the rotation mode, the automatic image rotation algorithm is used. This algorithm relies on and works best with images that have a good quality machine printed text. The algorithm does not fully work with images that contain nine-pin dot-matrix text or other non-machine printed text. Also, if an image contains text with various orientations, such as vertical and horizontal text, the image might be rotated in an undesirable way.

To avoid the possibility of the recognition engine not releasing the image until ruleset completion, call this action in a separate ruleset after recognition.

This action supports the automatic retry mechanism.

Example:

In the following example, the first instance of this action rotates the current page image file automatically and saves the applied rotation mode in the variable OCRSRotMod. The second instance applies a 90-degree rotation to the current image.

```
RotateImageExOCR_S ("", "")  
RotateImageExOCR_S ("", "2")
```

Parent topic: [OCR_SR actions](#)

RotateImageOCR_S

Use with the RotateTIO action from the Recog_Shared library to update the CCO file with the correct position coordinates after image rotation.

Member of namespace

OCR_SR

Syntax

```
bool RotateImageOCR_S ()
```

Parameters

None

Returns

False if the ruleset with this action is not bound to a Page object of the Document Hierarchy, or if the action cannot locate the image file representing the current page. Otherwise, True.

Level

Page only.

Details

This action performs automatic rotation of black and white .TIF (or .TIFF) files. The automatic image rotation algorithm relies on, and works best with, images with good quality machine printed text. If an image contains text with various orientations, for example vertical and horizontal, the image might be rotated undesirably. The automatic rotation algorithm does not fully work with images containing 9-pin dot-matrix text or other non-machine printed text.

It is recommended that this action be called in a separate ruleset after recognition due to instances where the recognition engine will not release the image until the ruleset has completed. This problem can manifest as a `cco does not exist` error in the log file.

This action supports the automatic retry mechanism.

Example

```
RotateImageOCR_S ()  
RecognizePageICR_C ()
```

In this example, automatic image rotation is performed prior to full page recognition via the ICR_C actions.

Parent topic: [OCR_SR actions](#)

Related reference:

[SetupAutomaticRetry](#)

SetContinueOnFailureOCR_S

Sets a value indicating whether a recognition action will abort or continue after it has encountered an error.

Member of namespace

OCR_SR

Syntax

```
bool SetContinueOnFailureOCR_S (bool ContinueOnFailure)
```

Parameters

ContinueOnFailure - A value indicating whether a recognition action will abort or continue after it has encountered an error. If this action is not called, the default value of True is used.

Returns

Always True.

Level

Page or Field.

Details

This action sets a value indicating whether a recognition action will abort or continue after it has encountered an error. If this action is not called, the default behavior is for the recognition actions to continue on error. This action is provided for legacy support of rules that manually perform a retry of recognition failures and is provided for backwards compatibility. It is recommended that you allow OCR_SR actions to automatically handle retries of recognition failures. When automatic mode is used, this action does not need to be called.

Example:

```
SetContinueOnFailureOCR_S(true)
```

Parent topic: [OCR_SR actions](#)

SetEngineTimeoutOCR_S

Specifies the number of seconds to wait before it is determined that an OCR/S recognition action is not running properly.

Important:

- If legacy mode is not currently enabled, calling this action enables legacy mode and disables automatic retry mode.
- In legacy mode, rules must be configured to retry recognition operations that fail. To avoid unnecessary rule configuration, use automatic retry instead of calling this action. For more information, see [SetupAutomaticRetry](#).

Member of namespace

OCR_SR

Syntax

```
bool SetEngineTimeoutOCR_S (int Seconds)
```

Parameters

Seconds
Type int

Parameters

Seconds: The value that indicates the number of seconds to wait before it is determined that an OCR/S recognition action is stalled or exited.

Returns

False, if the parameter is not numeric or is less than 1. Otherwise, True.

Level

Page or Field.

Details

This action sets the number of seconds to wait before it is assumed that an OCR/S recognition action is no longer running correctly. When the timeout is reached, the recognition process is removed from memory.

If a recognition action does not complete within the specified number of seconds indicated by a `SetOutOfProcessRecogTimeout` action or a `SetEngineTimeout` action, it is assumed that the recognition engine encountered a severe error. It is removed from memory, and recognition is automatically restarted one more time. If the recognition action completes successfully within the specified time on either the first or second attempt, that recognition action is successful. If the recognition action does not complete by the specified time on the second attempt, the recognition action is set to abort, if the `RecogContinueOnFailure(False)` action was used.

If `SetEngineTimeout` is not called, the default value of 180 seconds is used. In normal conditions, the default value is sufficient and does not need to be changed. This value must be increased only if a single page consistently takes more than 3 minutes to complete, which is not a typical situation. The programmer can choose to shorten this time to reduce the time to detect failures earlier, provided there is time to perform recognition in worst case scenarios. For best results, this timeout can be set the same or longer than the value specified in a `SetOutOfProcessRecogTimeout` action.

Example

```
SetEngineTimeoutOCR_S(180)
RecognizeFieldOCR_S
```

Parent topic: [OCR_SR actions](#)

SetOutOfProcessLoggingOCR_S

Writes logs for the out-of-process process for debugging.

Member of namespace

OCR_SR

Syntax

```
bool SetOutOfProcessLoggingOCR_S (bool EnableLogging)
```

Parameters

EnableLogging - A value indicating whether to write logs for the out-of-process process for debugging.

Returns

Always True.

Level

Page or Field.

Details

Writes logs for the out-of-process process for debugging. This log file is written to the batch folder. Multiple logs are created and they are named, `dcoprocessor.logdcoproc XX_XX_XX.log`, where X is numeric.

Example:

```
SetOutOfProcessLoggingOCR_S(true)
```

Parent topic: [OCR_SR actions](#)

SetOutOfProcessTimeoutOCR_S

Sets the number of seconds to wait to determine that an out-of-process action has stalled.

Member of namespace

OCR_SR

Syntax

```
bool SetOutOfProcessTimeoutOCR_S (int Seconds)
```

Parameters

Seconds - A value indicating the number of seconds to wait to determine that an out-of-process recognition action has stalled.

Returns

False, if the parameter is not numeric or less than 1. Otherwise, True.

Level

Page or Field.

Details

This action sets the number of seconds to wait to determine that an out-of-process action has stalled. If a recognition action does not complete within the specified number of seconds indicated by a *SetOutOfProcessTimeoutOCR_S* action or a *SetEngineTimeoutOCR_S* action, it is assumed that the recognition engine has encountered a severe error, and the recognition is removed from memory. A variable with *RecogStatus* of -1 is created in the calling object in this situation. If *SetOutOfProcessTimeoutOCR_S* is not called, the default value of 600 seconds is used. In normal scenario, the default value is sufficient and does not need to be changed. You may need to increase this value only if a single page consistently takes more than 10 minutes to complete, which is a very atypical situation.

Example:

```
SetOutOfProcessTimeoutOCR_S(180)
```

Parent topic: [OCR_SR actions](#)

UseOutOfProcessRecogOCR_S

Sets whether recognition will run out-of-process or in process. Running out of process improves stability of the application.

Member of namespace

OCR_SR

Syntax

```
bool UseOutOfProcessRecogOCR_S (bool UseOutOfProcessRecog)
```

Parameters

UseOutOfProcessRecog - A value indicating whether recognition will run out-of-process. Set to **True** for out-of-process recognition, and **False** for in-process recognition. By default, the recognition actions run out-of-process.

Returns

Always True.

Level

Page or Field.

Details

This action indicates whether recognition will run out-of-process or in process. Running out of process improves stability of the application.

Example:

```
UseOutOfProcessRecogOCR_S(true)
```

Parent topic: [OCR_SR actions](#)

OpenTextFaxServer actions

Use the OpenTextFaxServer actions to import faxes from an OpenTextFaxServer.

You can use the OpenTextFaxServer actions to create Datacap document batches from incoming faxes. You can also use these actions send the contents of a document to a specified fax number.

- [Connect](#)
Creates the connection to the Fax server.
- [ContinueOnConnectionError](#)
Specifies whether the batch should continue if there is an error connecting to the server.
- [ContinueOnFaxImportError](#)
Specifies whether the batch should abort if there is an error importing a fax.
- [Disconnect](#)
Disconnects the connection from the Fax server.

- [ImportFaxes](#)
Imports the faxes from the Fax server into the document batch.
- [SendAsFax](#)
Faxes the contents of the document or page to the specified Fax number.
- [SetAbortTimeout](#)
Sets the amount of time to wait before you stop running a batch.
- [SetFaxRemovalAfterImport](#)
Sets whether to remove processed faxes from the Fax server. This action must be set to True to enable the import of new faxes.
- [SetInputFolder](#)
Sets the name of the input folder where faxes are to be imported from.
- [SetMaxNumberOfFaxes](#)
Sets the maximum number of faxes that are allowed per batch.
- [SetNumberOfRetries](#)
Sets the number of times to attempt a connection to the Fax server after a connection error occurs.
- [SetPollingInterval](#)
Sets the number of milliseconds to wait before the OpenTextFaxServer resumes fax polling from the Fax server.
- [SetProcessedFaxesFolder](#)
Sets the name of the folder where faxes are to be moved to after they are imported.
- [SetProtocol](#)
Sets the protocol to use to connect to the Fax server.
- [SetRetryTimeout](#)
Sets the number milliseconds to wait before attempting a connection to the fax server after a connection error occurs.
- [SetServerName](#)
Sets the name of the Fax server to which you can upload faxes.
- [SetUserID](#)
Sets the user ID used to log in to the Fax server.
- [SetUserPassword](#)
Sets the password used to log in to the Fax server.
- [SetWindowsAuthentication](#)
Sets whether to use Windows Authentication to connect to the Fax server.

Parent topic: [Global actions](#)

Connect

Creates the connection to the Fax server.

Member of namespace

OpenTextFaxServer

Syntax

```
bool Connect ()
```

Returns

False if the action is not called at the batch level or if the connection to fax server cannot be established. Otherwise, True.

Level

Batch Level.

Details

Connects to the fax server. This action should be called after setting the server connection parameters via the following actions:

- `SetServerName("myserver")`
- `SetUserID("myuser")`
- `SetUserPassword("mypassword")`
- `SetProtocol(4)`
- `SetWindowsAuthentication(True)`

Example

```
SetServerName ("myserver")
SetUserID ("myuser")
SetUserPassword ("mypassword")
SetProtocol (4)
Connect ()
```

Parent topic: [OpenTextFaxServer actions](#)

ContinueOnConnectionError

Specifies whether the batch should continue if there is an error connecting to the server.

Member of namespace

OpenTextFaxServer

Syntax

```
bool ContinueOnConnectionError (bool Continue)
```

Parameters

Continue
Type: bool

Parameters

A boolean value specifying whether or not the batch should abort if there is an error connecting to the server.

Returns

Always True.

Level

Any level.

Details

When the parameter is set to True, the batch finishes with Pending status, avoiding the creation of more batches that will result in an aborted status.

If this action is not called, the default value of False is used and the batch is aborted at the end of processing.

Include this action before the Connect() action.

Example

```
ContinueOnConnectionError(true)
SetNumberOfRetries(3)
SetServerName("myserver")
SetWindowsAuthentication(True)
SetProtocol(4)
Connect()
ImportFaxes()
```

Parent topic: [OpenTextFaxServer actions](#)

ContinueOnFaxImportError

Specifies whether the batch should abort if there is an error importing a fax.

Member of namespace

OpenTextFaxServer

Syntax

```
bool ContinueOnFaxImportError (bool Continue)
```

Parameters

Continue
Type: bool

Parameters

A boolean value specifying whether or not the batch should abort if there is an error importing a fax.

Returns

Always True.

Level

Any level.

Details

Sets a boolean value specifying whether or not the batch should abort if there is an error importing a fax. When the parameter is set to True the batch finishes with Pending status, and contains all faxes that were imported

successfully, up to the last one that failed to be imported.

If `ContinueOnFaxImportError` is never called, the `ImportFaxes` action continue processing after an error. Call `ContinueOnFaxImportError(False)` to stop ingestion of faxes after an error.

Include this action before the `Connect()` action.

Example

```
ContinueOnFaxImportError (true)
SetNumberOfRetries (3)
SetServerName ("myserver")
SetWindowsAuthentication (True)
SetProtocol (4)
Connect ()
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

Disconnect

Disconnects the connection from the Fax server.

Member of namespace

`OpenTextFaxServer`

Syntax

```
bool Disconnect ()
```

Returns

False if the action is not called at the batch level or if the connection to fax server cannot be closed. Otherwise, True.

Level

Batch Level.

Details

Disconnects to the fax server. This action should be called after the `Import()` or `Connect()` actions. Typically this action would be called at the Batch's close node, after the connection to the fax server is made and faxes are imported.

Example

```
SetServerName ("myserver")
SetUserID ("myuser")
SetUserPassword ("mypassword")
SetProtocol (4)
Connect ()
ImportPages ()
Disconnect ()
```

Parent topic: [OpenTextFaxServer actions](#)

ImportFaxes

Imports the faxes from the Fax server into the document batch.

Member of namespace

OpenTextFaxServer

Syntax

```
bool ImportFaxes ()
```

Returns

False if the action is not called at the batch level or if an exception is encountered while importing faxes. Otherwise, True.

Level

Batch Level.

Details

This action imports faxes from the fax server. Each fax that is imported is stored in a document inside the Datacap batch. The following fax information will be stored in the document's variables (some of these variables can be empty):

- FaxUniqueID
- FaxStatus
- TotalPages
- LastHistoryChangeDateTime
- FromFaxNumber
- FromName
- FromVoiceNumber
- FromGeneralFaxNumber
- FromGeneralVoiceNumber
- Attachments
- ToFaxNumber
- ToVoiceNumber

Attention: Setting the batch variable *WriteFaxXMLData* to "1" causes the action to write all possible fax properties to an XML file. The XML is named based on the created document ID for a fax, for example, 20120109.000008.01.xml.

Include this action after a `Connect()` action.

Example

```
SetServerName ("myserver")
SetUserID ("myuser")
SetUserPassword ("mypassword")
SetProtocol (4)
Connect ()
ImportFaxes ()
```

Note: If the `Connect()` action is not called prior to calling `ImportFaxes()`, `ImportFaxes()` automatically calls the `Connect()` action. However, the actions that set the connection parameters need to be called prior to `ImportFaxes()`.

Parent topic: [OpenTextFaxServer actions](#)

SendAsFax

Faxes the contents of the document or page to the specified Fax number.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SendAsFax (string ToFaxNumber, string ToName)
```

Parameters

ToFaxNumber
Type: string
ToName
Type: string

Parameters

- ToFaxNumber: Recipient's fax number. This parameter is required. Smart parameters are supported.
- ToName: Recipient's name. This parameter is optional. If empty, the default ToName configured for the logged in user (on the server) is used. Smart parameters are supported.

Returns

False, if the action is not called at the document or page levels, or the fax number is not specified, or the document does not contain pages (attachments), or if a connection cannot be made to the fax server, or if the fax server returns an exception while attempting to send the fax. Otherwise, True.

Level

Document and Page levels.

Details

Faxes the document contents to the specified fax number.

Example

```
SendAsFax ("123-456-8971", "John Doe")
```

A connection to the Fax server must be established via actions before you can use the SendAsFax action.

Parent topic: [OpenTextFaxServer actions](#)

SetAbortTimeout

Sets the amount of time to wait before you stop running a batch.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetAbortTimeout (int Milliseconds)
```

Parameters

Milliseconds
Type: int

Parameters

Milliseconds : The amount of time, in milliseconds, to wait before aborting a batch. The default value is 10000 ms (10 seconds).

Returns

False if the action is not called at the batch level. Otherwise, True.

Level

Batch Level.

Details

Sets the amount of time to wait before aborting a batch.

The action waits the specified time before returning when an abort occurs. This action can be useful to prevent a large number of aborted batches due to an abort condition. For example, if the fax server should become unavailable for a time, the abort timeout will limit the amount of aborted batches until the fax server becomes available again.

If this action is not called, the default value of 10 seconds is used.

Include this action before a ImportFaxes() action.

Example

```
SetServerName ("myserver")
SetUserID ("myuser")
SetUserPassword ("mypassword")
SetProtocol (4)
SetAbortTimeout (5000)
Connect ()
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetFaxRemovalAfterImport

Sets whether to remove processed faxes from the Fax server. This action must be set to True to enable the import of new faxes.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetFaxRemovalAfterImport (bool RemoveFaxes)
```

Parameters

RemoveFaxes
Type: bool

Parameters

A boolean that sets whether or not to remove processed faxes from the server. The default value is False.

- True : Faxes will be removed from the fax server once they are imported into a Datacap batch.
- False : Faxes will remain in the fax server once they are imported into a Datacap batch.

The default value is False.

Returns

False if the action is not called at the batch level. Otherwise, True.

Level

Batch Level.

Details

Sets whether or not to remove processed faxes from the server after they have been imported into the Datacap batch.

If this action is not called, the default value of False is used.

Include this action before a ImportFaxes() action.

Example

```
SetServerName ("myserver")
SetUserID ("myuser")
SetUserPassword ("mypassword")
SetProtocol (4)
SetFaxRemovalAfterImport (True)
Connect ()
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetInputFolder

Sets the name of the input folder where faxes are to be imported from.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetInputFolder (string FolderName)
```

Parameters

FolderName
Type: string

Parameters

An string value representing the name of the user folder where faxes are to be imported from.

Returns

Always True.

Level

Any level.

Details

If this action is not called, the faxes are imported from the default user folder.

Include this action before a `ImportFaxes()`.

Example

```
SetNumberOfRetries (3) >  
SetRetryTimeout (3000)  
SetServerName ("myserver")  
SetWindowsAuthentication (True)  
SetProtocol (4)  
Connect ()  
SetInputFolder (INPUT)  
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetMaxNumberOfFaxes

Sets the maximum number of faxes that are allowed per batch.

Member of namespace

Syntax

```
bool SetMaxNumberOfFaxes (int MaxFaxes)
```

Parameters

MaxFaxes
Type: int

Parameters

MaxFaxes : The maximum number of faxes allowed per batch. The default value is 100.

Returns

False if the action is not called at the batch level. Otherwise, True.

Level

Batch Level.

Details

Sets the maximum number of faxes allowed per batch.

If this action is not called, the default value of 100 faxes per batch is used.

Include this action before a ImportFaxes() action.

Example

```
SetServerName ("myserver")  
SetUserID ("myuser")  
SetUserPassword ("mypassword")  
SetProtocol (4)  
SetMaxNumberOfFaxes (5)  
Connect ()  
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetNumberOfRetries

Sets the number of times to attempt a connection to the Fax server after a connection error occurs.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetNumberOfRetries (int NumberOfRetries)
```

Parameters

NumberOfRetries
Type: int

Parameters

An integer value representing the number of times to attempt a connection to the fax server after a connection error occurs.

Returns

Always True.

Level

Any level.

Details

Sets the number of times to attempt a connection to the fax server after a connection error occurs.

If this action is not called, the default value of 3 is used.

Include this action before the Connect() action.

Example

```
SetNumberOfRetries (3)
SetServerName ("myserver")
SetWindowsAuthentication (True)
SetProtocol (4)
Connect ()
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetPollingInterval

Sets the number of milliseconds to wait before the OpenTextFaxServer resumes fax polling from the Fax server.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetPollingInterval (int Milliseconds)
```

Parameters

Milliseconds
Type: int

Parameters

Milliseconds : The amount of time, in milliseconds, to wait before polling the fax server again. The default value is 2000 ms (2 seconds).

Returns

False if the action is not called at the batch level. Otherwise, True.

Level

Batch Level.

Details

Sets the amount of time to wait before resuming fax polling from the server.

If this action is not called, the default value of 2 seconds is used.

Include this action before a `ImportFaxes()` action.

Example

```
SetServerName ("myserver")
SetUserID ("myuser")
SetUserPassword ("mypassword")
SetProtocol (4)
SetPollingInterval (5000)
Connect ()
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetProcessedFaxesFolder

Sets the name of the folder where faxes are to be moved to after they are imported.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetProcessedFaxesFolder (string FolderName)
```

Parameters

FolderName
Type: string

Parameters

A string value that represents the name of the user folder where faxes are to be moved to after they are imported.

Returns

Always True.

Level

Any level.

Details

If this action is not called, the faxes remain in the input folder.

Include this action before a `ImportFaxes()`.

Example

```
SetNumberOfRetries (3) >
SetRetryTimeout (3000)
SetServerName ("myserver")
SetWindowsAuthentication (True)
SetProtocol (4)
Connect ()
SetProcessedFaxesFolder (OUTPUT)
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetProtocol

Sets the protocol to use to connect to the Fax server.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetProtocol (int Protocol)
```

Parameters

Protocol
Type: int

Parameters

The protocol to be used to connect to the fax server. The default value is 4 (TCPIP).
Valid parameter values are:

- 1 : Named Pipes
- 2 : IPXOS2
- 3 : SPX
- 4 : TCPIP
- 5 : IPX

- 6 : SecTCPIP
- 7 : SecSPX

Returns

False if the action is not called at the batch level or if the parameter is invalid. Otherwise, True.

Level

Batch Level.

Details

Sets the protocol to be used to connect to the fax server.

If this action is not called, the default value of 4 (TCPIP protocol) is used.

Include this action before a `ImportFaxes()` or `Connect()` action.

Example

```
SetServerName ("myserver")
SetUserID ("myuser")
SetUserPassword ("mypassword")
SetProtocol (4)
Connect ()
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetRetryTimeout

Sets the number milliseconds to wait before attempting a connection to the fax server after a connection error occurs.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetRetryTimeout (int Milliseconds)
```

Parameters

Milliseconds
Type: int

Parameters

An integer value representing the number milliseconds to wait before attempting a connection to the fax server after a connection error occurs.

Returns

Always True.

Level

Any level.

Details

If this action is not called, the default value of 3000 milliseconds is used.

Include this action before a `ImportFaxes()` or `Connect()` action.

Example

```
SetNumberOfRetries (3) >  
SetRetryTimeout (3000)  
SetServerName ("myserver")  
SetWindowsAuthentication (True)  
SetProtocol (4)  
Connect ()  
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetServerName

Sets the name of the Fax server to which you can upload faxes.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetServerName (string ServerName)
```

Parameters

ServerName
Type: string

Parameters

ServerName : The name of the fax server. Smart parameters are supported.

Returns

False if the action is not called at the batch level. Otherwise, True.

Level

Batch Level.

Details

Sets the name of the fax server to connect to.

Include this action before an ImportFaxes() or Connect() action

Example

```
SetServerName ("myserver")
SetUserID ("myuser")
SetUserPassword ("mypassword")
SetProtocol (4)
Connect ()
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetUserID

Sets the user ID used to log in to the Fax server.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetUserID (string UserID)
```

Parameters

UserID
Type: string

Parameters

UserID : The user ID to be used to connect to the fax server. Smart parameters are supported.

Returns

False if the action is not called at the batch level. Otherwise, True.

Level

Batch Level.

Details

Sets the user ID to connect to the fax server.

Include this action before a ImportFaxes() or Connect() action.

Example

```
SetServerName ("myserver")
SetUserID ("myuser")
SetUserPassword ("mypassword")
```

```
SetProtocol (4)
Connect ()
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetUserPassword

Sets the password used to log in to the Fax server.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetUserPassword (string UserPassword)
```

Parameters

UserPassword
Type: string

Parameters

UserPassword : The user ID password to connect to the fax server. Smart parameters are supported.

Returns

False if the action is not called at the batch level. Otherwise, True.

Level

Batch Level.

Details

Sets the user ID password to connect to the fax server.

Include this action before a `ImportFaxes()` or `Connect()` action.

It is recommended that you create an advanced value in the custom values tab in the Application Manager to encrypt your password instead of hard coding it in the action parameter. The password can be retrieved using smart parameters.

Example

```
SetServerName ("myserver")
SetUserID ("myuser")
SetUserPassword ("mypassword")
SetProtocol (4)
Connect ()
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

SetWindowsAuthentication

Sets whether to use Windows Authentication to connect to the Fax server.

Member of namespace

OpenTextFaxServer

Syntax

```
bool SetWindowsAuthentication (bool UseWindowsAuthentication)
```

Parameters

UseWindowsAuthentication
Type: bool

Parameters

Sets whether or not to use Windows Authentication to connect to the fax server. The default value is False.

- True : Windows Authentication will be used. The actions SetUserID() and SetUserPassword() are not required when UseWindowsAuthentication is set to True.
- False : Fax Server user authentication will be used. The actions SetUserID() and SetUserPassword() are required when UseWindowsAuthentication is set to False.

The default value is False.

Returns

False if the action is not called at the batch level. Otherwise, True.

Level

Batch Level.

Details

Include this action before a ImportFaxes() or Connect() action.

Example

```
SetServerName ("myserver")
SetWindowsAuthentication (True)
SetProtocol (4)
Connect ()
ImportFaxes ()
```

Parent topic: [OpenTextFaxServer actions](#)

PatternMatch actions

Use the `PatternMatch` actions for pattern-based page identification and for page registration (alignment). Page registration is important when you are working with OMR check boxes.

`PatternMatch` actions look for a match to specified anchor patterns, identifies the page, sets the page type, and sets the confidence level for pattern matching.

- [MatchPattern](#)
Align the image of this field on the current page with the fingerprint
- [pat_RecogMatch_Id](#)
Identifies the current page type by matching OCR results in any fingerprint Anchor zone with OCR results for the corresponding zone on the current page.
- [pat_RegisterZones](#)
Registers and adjusts the positions of all fields on the current source page, based on the positions of the page's designated Anchor field(s).
- [pat_ReleasePageAnchors](#)
An action that can be called at the end of a batch to release information about the identity and location of a page's Anchor field(s).
- [PatternMatch_Fingerprint](#)
Identifies a page from a specified list of fingerprints.
- [PatternMatch_Identify](#)
Identifies a page by using image pattern matching.
- [PatternMatch_PageType](#)
Identifies a page according to its Page Type.
- [SetMatchConfidence](#)
Sets the confidence threshold for pattern matching.

Parent topic: [Global actions](#)

MatchPattern

Align the image of this field on the current page with the fingerprint

Member of namespace

`PatternMatch`

Syntax

```
bool MatchPattern ()
```

Parameters

None.

Returns

False, if any of the follow conditions occur.

- The Anchor position that is returned is not Numeric.
- No image is found.
- The accuracy of the match is below the set Confidence Value.
- An Anchor match does not occur.

Otherwise, True.

Level

Field level.

Details

Searches on the current image in a zone that is associated with the current field for a match to the pattern specified for this field in a fingerprint. The zoned area from the original fingerprint is matched against a larger zone in the current image. The search area is controlled by the METRIC variable. METRIC=200,100 means search from 200 pixels to the left and right, and 100 pixels above and below the expected location. If METRIC is not specified, the default is 500 pixels horizontal and vertical.

The fingerprint is determined by the current image's Fingerprint ID, or the Global Fingerprint ID if the current image is not identified.

MatchPattern can be called on any field and if matched, an offset variable is saved for that field. If called before ReadZones, then ReadZones uses the offset for that field when its position is set. Other fields are unaffected.

If the field is matched to the fingerprint with a confidence equal to or greater than the required confidence, the position of the field is set to the found location. If the field is not matched, the function returns false. If not found and the field's Required variable is non-zero, the field status is set to 1 (Error or Validation failed).

This action operates on black and white images, grayscale or color images cause the action to fail. The fingerprint image must have the same resolution (DPI) as the current page image. The geometric shape that is contained in the Anchor field must be bold and well-defined with clear edges, with crisp, black and white markings that product a distinct shape. The shape must be thick and compact, not composed of long thin lines. To avoid false positive matches, the shape must not match other shapes or black areas that might exist nearby within the same image.

Example

```
MatchPattern ()
```

Parent topic: [PatternMatch actions](#)

pat_RecogMatch_Id

Identifies the current page type by matching OCR results in any fingerprint Anchor zone with OCR results for the corresponding zone on the current page.

Member of namespace

PatternMatch

Syntax

```
bool pat_RecogMatch_Id ()
```

Parameters

None.

Returns

True if the ruleset is bound to a Page, and a fingerprint matching the text of at least one Anchor field is found. Otherwise, and in case of any errors, False. In addition, the page variable *TemplateID* is set to the matching Fingerprint ID.

Level

Page only.

Details

pat_RecogMatch_Id identifies the page by matching text from any fingerprint Anchor field with the corresponding text on the current page. Fuzzy matching is used. Full-page OCR or ICR must be performed prior to calling this action.

If any Anchor field text in the current page matches the zonal text of any fingerprint, the page is identified by that fingerprint (first match). The Type of the current page is set to the fingerprint page type if a match is found. Full page OCR or ICR must be performed on both the fingerprints and the current image prior to calling this action. Text to be matched is extracted from each fingerprint's Anchor field, which should be defined tightly around the text in the fingerprint. The search area in the current image is the fingerprint-specific field zone in the Document Hierarchy, extended by any associated METRIC variable.

Page identification using pat_RecogMatch_ID (text matching) is mutually exclusive with identification using graphical pattern matching actions (PatternMatch_Identify, etc.). Anchor fields in the Document Hierarchy should be selected carefully so that false positive text matches do not occur.

Example

```
pat_RecogMatch_Id()
```

Parent topic: [PatternMatch actions](#)

pat_RegisterZones

Registers and adjusts the positions of all fields on the current source page, based on the positions of the page's designated Anchor field(s).

Member of namespace

PatternMatch

Syntax

```
bool pat_RegisterZones ()
```

Parameters

None.

Returns

True if the ruleset with this action is bound to a Page object of the Document Hierarchy, and if the action can find all designated Anchor fields. Otherwise, False.

Level

Page only.

Details

`pat_RegisterZones` registers and adjusts the positions of all fields on the current page, based on the previously matched positions of the page's designated Anchor field(s). Anchor fields are determined by the Anchor Field setting in Datacap Studio, for each field. Prior to calling `pat_RegisterZones`, usually in a different task or ruleset, one of the `PatternMatch` actions that performs Anchor matching must be called. Then, when the `pat_RegisterZones` action is called, the expected positions of the Anchor fields on the image (taking into account the Fingerprint classification) are compared with the recognized positions of the fields identified as an Anchor field. The action `ReadZones` must be called prior to `pat_RegisterZones`. If any required Anchors are not matched, an operator may be required to update the Anchor position in a verify or fixup task. All matched or manually adjusted Anchor positions are used for adjustment, Anchors that are not matched are ignored.

- If one Anchor is found, the field positions are all shifted by the same amount.
- If two or more Anchors are found, the field positions are shifted by different amounts, depending on their distance from each Anchor. This process is called Interpolation.

The expected positions of the Anchor fields on the image (taking into account the Fingerprint classification) are compared with the recognized positions of those Anchor fields - or the Anchor positions set manually by a Fixup task's operator.

Example

```
ReadZones ()
pat_RegisterZones ()
PrecognizePageFieldsOCR_S ()
```

Parent topic: [PatternMatch actions](#)

pat_ReleasePageAnchors

An action that can be called at the end of a batch to release information about the identity and location of a page's Anchor field(s).

Member of namespace

PatternMatch

Syntax

```
bool pat_ReleasePageAnchors ()
```

Parameters

None.

Returns

Always True.

Level

Page only.

Details

This action can be optionally called to release the small amount of Anchor memory that was allocated by the action `pat_RegisterZones`. If `pat_ReleasePageAnchors` is not called, the memory will be released at the end of the batch or the next time `pat_RegisterZones` is called.

Example

```
pat_ReleasePageAnchors ()
```

Parent topic: [PatternMatch actions](#)

PatternMatch_Fingerprint

Identifies a page from a specified list of fingerprints.

Member of namespace

PatternMatch

Syntax

```
bool PatternMatch_Fingerprint (StrParam)
```

Parameters

A comma-separated list of one or more Fingerprint IDs.

Returns

False, if the rule that contains this action was not applied to a Page object of the Document Hierarchy; if a parameter is invalid; if a match does not occur; or if one or more of the specified fingerprints do not exist. Otherwise, True.

Level

Page level only.

Details

`PatternMatch_Fingerprint` identifies a page's type and fingerprint by using geometric pattern matching. The locations of unique patterns are configured as Anchor fields for each fingerprint in the Document Hierarchy. One or more Anchor fields can be used to match geometric shapes on a fingerprint to the current image. If one or more Anchor fields on the current page match a fingerprint with equal to or greater than the configured confidence level, the page is identified with that fingerprint. The action does not require all defined anchors to match - the first match is used. The action loads all Anchor field patterns from the specified fingerprints, then searches on the current image for each of the patterns in the associated zones. The search area for each zone is increased by the dimensions that are specified in the page METRIC variable. If METRIC is not specified, the default is 500 pixels horizontal and vertical. When this action finds a match, it sets the matching Fingerprint ID

and Page Type. It also creates page-level fields and update the Anchor fields with Anchor-specific pattern offset values in a field-level *Image_Offset* variable. The offset can be used subsequent to matching a fingerprint. The `pat_RegisterZones` action can be used to align the zones in the fingerprint to the current image, providing more accurately positioned text in each field.

This action requires the current page image to be bi-tonal (black and white). Grayscale or color images cause the action to fail. The fingerprint image must have the same resolution (DPI) as the current page image. The geometric shape that is contained in each Anchor field must be bold and well-defined with clear edges, with crisp, black and white markings that produce a distinct shape. The shape must be thick and compact, not composed of long thin lines. To avoid false positive matches, the shape must not match other shapes or black areas that might exist nearby within the same image.

Example

```
PatternMatch_Fingerprint(1024,1034,1035,1036)
```

This example compares the current page to the four fingerprints that are specified by their IDs.

Parent topic: [PatternMatch actions](#)

PatternMatch_Identify

Identifies a page by using image pattern matching.

Member of namespace

PatternMatch

Syntax

```
bool PatternMatch_Identify ()
```

Parameters

None.

Returns

False, if the rule that contains this action was not applied to a Page object of the Document Hierarchy; if a pattern match is not found; or if fingerprints do not exist. Otherwise, True.

Level

Page level only.

Details

`PatternMatch_Identify` identifies a page's type and fingerprint by using geometric pattern matching. The locations of unique patterns are configured as Anchor fields for each fingerprint in the Document Hierarchy. One or more Anchor fields can be used to match geometric shapes on a fingerprint to the current image. If one or more Anchor fields on the current page match a fingerprint with equal to or greater than the configured confidence level, the page is identified with that fingerprint. The action does not require all defined anchors to match - the first match is used. The action loads all Anchor field patterns from the fingerprint library, then searches on the current image for each of the patterns in the associated zones. The search area for each zone

is increased by the dimensions that are specified in the page METRIC variable. If METRIC is not specified, the default is 500 pixels horizontal and vertical. When this action finds a match, it sets the matching Fingerprint ID and Page Type. It also creates page-level fields and update the Anchor fields with Anchor-specific pattern offset values in a field-level *Image_Offset* variable. The offset can be used subsequent to matching a fingerprint. The *pat_RegisterZones* action can be used to align the zones in the fingerprint to the current image, providing more accurately positioned text in each field.

This action requires the current page image to be bi-tonal (black and white). Grayscale or color images cause the action to fail. The fingerprint image must have the same resolution (DPI) as the current page image. The geometric shape that is contained in each Anchor field must be bold and well-defined with clear edges, with crisp, black and white markings that produce a distinct shape. The shape must be thick and compact, not composed of long thin lines. To avoid false positive matches, the shape must not match other shapes or black areas that might exist nearby within the same image.

Example

```
PatternMatch_Identify()
```

Parent topic: [PatternMatch actions](#)

PatternMatch_PageType

Identifies a page according to its Page Type.

Member of namespace

PatternMatch

Syntax

```
bool PatternMatch_PageType (StrParam)
```

Parameters

One or more Page Types defined in the Document Hierarchy

Returns

False if the rule containing this action was not applied to a Page object of the Document Hierarchy; if the parameter is invalid; if a match does not occur; or if fingerprints do not yet exist. Otherwise, True.

Level

Page level only.

Details

PatternMatch_PageType identifies a page's type and fingerprint using geometric pattern matching. The locations of unique patterns are configured as Anchor Fields for each fingerprint in the Document Hierarchy. One or more Anchor fields can be used to match geometric shapes on a fingerprint to the current image. If one or more Anchor fields on the current page match a fingerprint, at or above the configured confidence level, the page is identified with that fingerprint. The action does not require all defined anchors to match - the first match is used. The action loads all Anchor field patterns from fingerprints with the specified page types, then

searches on the current image for each of the patterns in the associated zones. The search area for each zone is increased by the dimensions specified in the page METRIC variable. If METRIC is not specified, the default is 500 pixels horizontal and vertical. When this action finds a match, it sets the matching Fingerprint ID and Page Type. It will also create page-level fields and update the Anchor fields with Anchor-specific pattern offset values in a field-level *Image_Offset* variable. The offset can be used subsequent to matching a fingerprint. The `pat_RegisterZones` action can be used to align the zones in the fingerprint to the current image, providing more accurately positioned text in each field.

This action requires the current page image to be bitonal (black and white), grayscale or color images will cause the action to fail. The fingerprint image must have the same resolution (DPI) as the current page image. The geometric shape contained in each Anchor field should be bold and well defined with clear edges, with crisp black and white markings, producing a distinct shape. The shape should be thick and compact, not composed of long thin lines. To avoid false positive matches, the shape should not match other shapes or black areas that may exist nearby within the same image.

Example

```
PatternMatch_PageType (HCFA 1500)
```

This action looks for a match among the inventory of fingerprints that have a page type of "HCFA 1500".

Parent topic: [PatternMatch actions](#)

SetMatchConfidence

Sets the confidence threshold for pattern matching.

Member of namespace

PatternMatch

Syntax

```
bool SetMatchConfidence (StrParam)
```

Parameters

The value of the confidence threshold.

The value must be between 0 (lowest confidence) and 9 (highest confidence).

Higher values require fewer differences between the compared areas to return a positive match value.

Returns

False if the parameter is not a number between 0 and 9. Otherwise, True.

Level

All.

Details

Sets the confidence threshold for pattern matching.

Example

```
SetMatchConfidence (9)
```

Parent topic: [PatternMatch actions](#)

Picture actions

Use the Picture actions to do field validations by picture strings. Picture strings define the supported format of a field such as a social security number, phone number, date.

A social security number, for example, is always *<three digits >-<two digits >-<four digits>*. You can define a picture string to represent this format and then use it to make sure that social security number fields contain conforming values.

- [PIC_ApplyPictureString](#)
Validates the current field by using the specified picture string.
- [PIC_FilterFields](#)
Validates the format of the current field, when called from a field, or all fields on the current page, when called from a page. Uses the picture string that is stored in the PictureString variable of the field.
- [PIC_FormatFields](#)
Validates the format of the current field or all fields on the current page and uses another to replace problem characters.
- [PIC_ReplaceBlankField](#)
If the current field is blank, sets the field value to the character that is specified.
- [PIC_SetPictureCharacter](#)
Defines up to 10 custom picture strings (0-9) that you can reference from the PIC_ApplyPictureString action.
- [PIC_ValidateField](#)
Validates the format of the current field by using the picture string that is stored in the PictureString variable of the field.

Parent topic: [Global actions](#)

PIC_ApplyPictureString

Validates the current field by using the specified picture string.

Syntax

```
bool PIC_ApplyPictureString (StrParam)
```

Parameters

The picture string to validate the field.

Returns

False, if called at the wrong level, if the picture string is longer than the field value or if the field fails the picture string validation. Otherwise, True.

Level

Field level.

Details

Validates the current field using a runtime *PictureString* as an argument. See the `PIC_FormatFields` action for picture string details.

Using the provided picture string, this action will test that each of the characters in the current field are allowed. The provided picture string must be the same length or shorter than the data on the field. If the picture string is shorter, then the last character of the picture string will be used to validate all remaining characters in the field.

See the help for action `PIC_FormatFields` for an overview of picture strings. Unlike `PIC_FormatFields` which uses the *PictureString* variable, `PIC_ApplyPictureString` accepts the picture string as a variable only.

Example

```
PIC_SetPictureCharacter("0,01")
PIC_SetPictureCharacter("1,0123")
PIC_SetPictureCharacter("2,-./")
PIC_ApplyPictureString("0N21N2NN")
```

This example creates custom picture strings for 0, 1 and 2. They are then used here to provide tighter control on the allowed input. "0N21N2NN" format matches a typical 6 digit date specification like "01/07/67".

Parent topic: [Picture actions](#)

PIC_FilterFields

Validates the format of the current field, when called from a field, or all fields on the current page, when called from a page. Uses the picture string that is stored in the *PictureString* variable of the field.

Syntax

```
bool PIC_FilterFields ()
```

Parameters

None.

Returns

Always True.

Level

All levels.

Details

Replaces a character and adjusts the confidence based on the *PictureString* defined for the field.

Lowers the Confidence Rating of any character in a field that does not satisfy the Picture String's criteria and replaces the problem character with a low confidence space. It is very similar to the `FormatFields` action but

does not use alternative recognition characters.

This action has two roles. If a character in the field does not match the picture string format defined for that field

1. It replaces any "problem" characters with a space character and marked as low confidence.
2. It lowers the Confidence Rating of any character in a field that does not satisfy the Picture String's criteria.
3. Any alternative recognition characters are removed from the field after execution.

Note: This Action is recursive and will affect all child fields of the calling node. While not direct input to this action, this action works with picture strings that are defined for a field. See the `PIC_FormatFields` action for a list of all available picture string codes and information about the `PictureString` variable.

Example

```
rrSet("XxN,@F.PictureString")
PIC_FilterFields()
```

This example expects the current field to have the first character be either an alphabetic character or a digit, the second character can be an alphabetic character, digit or punctuation character and the remaining characters must only be digits.

Parent topic: [Picture actions](#)

Related reference:

[PIC_FormatFields](#)

[PIC_ValidateField](#)

PIC_FormatFields

Validates the format of the current field or all fields on the current page and uses another to replace problem characters.

Syntax

```
bool PIC_FormatFields ()
```

Parameters

None.

Returns

Always True.

Level

All levels.

Details

Lowers the Confidence Rating of any character in a field that does not satisfy the `PictureString` criteria.

This action adjusts the character confidence of a field, and optionally replaces characters that are based on the picture string set for the field. It is similar to the `FilterFields` action.

This action has two roles, if a character in the field does not match the picture string format that is defined for that field.

1. It replaces any problem characters with an alternative character from a secondary recognition engine, if one exists. If an alternative recognition character does not exist, then the original character is unchanged.
Attention: There must be an equal number of alternative characters as the field length, and the alternative character must also be valid within the field's picture string for substitution to occur. If the alternative recognition character is also not a valid picture string character, then no substitution occurs.
2. It lowers the Confidence Rating of any character in a field that does not satisfy the picture string's criteria.

Alternative recognition characters are removed from the field after execution.

Important: This Action is recursive and affects all child fields of the calling node.

Picture Strings:

This action works with picture strings that are defined for a field. The picture string must be stored in a field variable called *PictureString*. Picture strings improve and filter recognition results, and are used to limit characters that are typed into that field during verify. The PIC_FormatFields and PIC_FilterFields actions can be called to enforce *PictureString* after recognition rules are started. The PIC_ApplyPictureString action is an exception that does not use the *PictureString* variable.

Recognition actions do not pay attention to this property. Individual recognition engines have their own parameters to help guide the recognition. The Web Verify task always enforces *PictureString*. Thick client Verify panels that are constructed by Batch Pilot Autoform also enforces *PictureString* specifications.

PIC_FilterFields replaces non-matching characters with low confidence spaces. PIC_FormatFields lowers the confidence. The DCEdit control enforces them during verify.

The picture values can be set in *PictureString* in two ways.

1. Use the rrunner action rrSet in a rule set. With this action, you can specify the *PictureString* variable and set it to the value that you want.
2. In the Zones tab of Datacap Studio, right-click on the field you want and choose Manage Variables.

While not direct input to this action as a standard parameter, here are the valid picture string characters that can be set in the *PictureString* field variable and are then used by this action.

- A: Alphabetic characters only or a space. Numeric and punctuation characters are not valid.
- a: Alphabetic, space and punctuation characters.
- D: Dates. The dates must be expressed with numeric characters. You can delimit months, dates, and years with hyphens, periods, and forward slashes.
- F: Float numbers, which are fractional numbers. To accommodate fractional values, you can include both numbers and a period (for the decimal separator) in this picture string. The F character allows minus signs to represent negative numbers.
- f: Numeric and punctuation characters.
- L: Lowercase alphabetic and space characters.
- l: Lowercase alphabetic, space, and punctuation characters.
- N: Numeric characters only.
- n: Uppercase alphabetic, numeric, or space characters.
- P: Punctuation and space characters.
- T: Time values. These values are expressed in numbers with a colon. In addition, the characters P, M, and A are allowed to distinguish between morning and afternoon times, and colon characters are allowed to delimit hours, minutes, and seconds.
- U: Uppercase alphabetic and space characters.

- u: Uppercase alphabetic, space, and punctuation characters.
- X: Alphabetic, space and numeric characters.
- x: Alphabetic, space, numeric, and punctuation characters.
- Z: Any character.
- #: Numeric characters and the minus sign.

PIC_SetPictureCharacter can be used to define up to 10 more application-specific picture strings at run time, identified as 0 through 9.

Example

```
rrSet("AN,@F.PictureString")
PIC_FormatFields()
```

This example expects the current field to contain a single alphabetic character followed by an unlimited number of digits. Here the *PictureString* variable is set at run time, but it can instead be configured at design time in the setup DCO in Datacap Studio.

Parent topic: [Picture actions](#)

Related reference:

[PIC_FilterFields](#)

[PIC_ValidateField](#)

[PIC_SetPictureCharacter](#)

PIC_ReplaceBlankField

If the current field is blank, sets the field value to the character that is specified.

Syntax

```
bool PIC_ReplaceBlankField (StrParam)
```

Parameters

A character or string that will be placed into the field if it is blank.

Returns

False if it is called at the wrong level or if the parameter is missing, otherwise True.

Level

Field level.

Details

- If a field is blank, it replaces it with a single character.
- If a field is empty or only contains spaces, it is replaced with the character or string that is passed in as a parameter.
- If the field is replaced with the input parameter, the confidence is changed to a low confidence of 1.

Example

```
PIC_ReplaceBlankField("~")
```

PIC_SetPictureCharacter

Defines up to 10 custom picture strings (0-9) that you can reference from the PIC_ApplyPictureString action.

Syntax

```
bool PIC_SetPictureCharacter (StrParam)
```

Parameters

Two comma-separated parameters

1. The picture string identifier. The value must be between 0 through 9.
2. A string of characters to associate with the picture string identifier (the first parameter).

Returns

False, if the parameter input is invalid. Otherwise, True.

Level

Any level.

Details

Configures application-specific picture strings.

In addition to the predefined character strings, custom picture strings can be configured. The picture string values 0 - 9 can be configured to allow validations that are not covered by the predefined settings.

It is possible to configure your verify panel edit control to restrict keyboard entry that is based on picture strings when you use the *PictureString* field variable. Only predefined picture strings work with the edit control. Any custom picture strings that are created by PIC_SetPictureCharacater action do not cause the edit control to restrict user input.

Examples

This example creates custom picture strings for 0, 1 and 2. They are then used to provide tighter control on the allowed input. "0N21N2NN" format matches a typical 6-digit date specification like "01/07/67".

```
PIC_SetPictureCharacter("0,01")
PIC_SetPictureCharacter("1,0123")
PIC_SetPictureCharacter("2,-./")
PIC_ApplyPictureString("0N21N2NN")
```

This example is the same except that the picture string is set up in the *PictureString* variable in Datacap Studio, so it is not seen here.

```
PIC_SetPictureCharacter("0,01")
PIC_SetPictureCharacter("1,0123")
PIC_SetPictureCharacter("2,-./")
PIC_ValidateField()
```

Related reference:[PIC_FormatFields](#)[PIC_FilterFields](#)[PIC_ApplyPictureString](#)

PIC_ValidateField

Validates the format of the current field by using the picture string that is stored in the `PictureString` variable of the field.

Syntax

```
bool PIC_ValidateField ()
```

Parameters

None.

Returns

False if the field value does not satisfy the Picture String criteria of the field. Otherwise True.

Level

Field level only.

Details

Checks the value of all characters in a field against that field's *PictureString* criteria.

If a character in the field does not match the picture string format defined for that field, it lowers the Confidence Rating of any character in a field that does not satisfy the Picture String's criteria. The criteria is stored in the *PictureString* variable that is bound to the field.

Note: Fields with a status of '-1' (Hidden) are checked but this action will not return false if the value does not match the picture string criteria.

While not direct input parameters to this action, this action works with picture strings that are defined for a field. See the `PIC_FormatFields` action for a list of all available picture string codes and information about the *PictureString* variable.

Example

```
PIC_ValidateField()
```

Parent topic: [Picture actions](#)

Related reference:[PIC_FormatFields](#)[PIC_FilterFields](#)[PIC_SetPictureCharacter](#)

POLR actions

Use the POLR action matches line items from your invoice image to the corresponding purchase order.

The POLR action pre-matches invoice line items with the purchase order before it runs the verification task.

- [CallPOLR](#)
Pre-matches invoice line items with the purchase order before it runs the verification task.

Parent topic: [Global actions](#)

CallPOLR

Pre-matches invoice line items with the purchase order before it runs the verification task.

Syntax

```
bool CallPOLR (StrParam)
```

Parameters

The ADOBDB constant number for the PO number field. When using bind variables, the data type of the PO Number is specified with this action.

Returns

Always True.

Level

Page level.

Details

This action is used to pre-match invoice line items with the PO prior to verify operator verification. The record set for the PO is retrieved using the information in the settings.ini. This calls the record set according to the DSN in the settings.ini. PODSN and POLookup indicate how to obtain the record set. The record set is expected to be the line items for the PO for the current document and is keyed off of the PO number. It then uses the POLR logic to perform the automatic matching.

Note: The TestPODSN and PODSN ini entries support smart parameters to allow for secure connection strings. The settings ini file must contain values for these keys:

- [POLR]
- Qty=
- ItemID=
- Price=
- WriteUnusedPOLInes=
- PriceTolerance=
- SeparatorCharacter=

If the station name has a suffix of "-Test" then this key must exist:

- [Database]
- TestPODSN=
- TestPOLookup=

If the station name does not have a suffix of "-Test" then this key must exist:

- [Database]
- PODSN=
- POLookup=

Example:

```
CallPOLR("200")
```

Parent topic: [POLR actions](#)

Recog_Shared actions

Use the Recog_Shared actions to do various fingerprint and recognition-related functions.

The Recog_Shared actions can recognize things like check box options and write the recognition results to the page data files.

- [AnalyzeImage](#)
Converts the Image file (.tif) that represents the current page to a Fingerprint file (.cco) file for the page.
- [CCONormalization_OFF](#)
Prevents the automatic running of NormalizeCCO procedures after a full-page recognition action was run.
- [CreateTextFile](#)
Creates a Text file (.txt) for the current page; adds the page's recognized values to the file; and places the file in the current batch, in your application's Batches directory.
- [IsBlankPage](#)
Counts the number of words in the Fingerprint file (.cco) file of the current page and returns True if the count is less than or equal to the number you enter as the parameter.
- [RecogContinueOnFailure](#)
Determines if a batch will abort if page or field recognition fails.
- [RecogOMRThreshold](#)
Performs OMR check box recognition by counting black pixels within each OMR box area in a Field with one or more OMR boxes.
- [RegisterPageFields](#)
Returns the field positions for all zoned fields of the current page.
- [RotateTio](#)
Checks if an Image file processed by the ImageFix action that assigns a .tio extension to the file needs to be rotated by 90, 180, or 270 degrees. If so, the action rotates and then saves the Image file with the same .tio extension.
- [SetAdjustFieldToChars](#)
Optional setting for SnapCCOToDCO to adjust the field position to its character positions.
- [SetFingerprintRecogPriority](#)
Sets the option that controls whether a full-page recognition action is to create a Fingerprint file(.cco) - aka a CCO file - for the current page.
- [SetFullPageRecogArea](#)
An optional action that sets the area of the current page that is to be the target of recognition procedures, when full-page recognition action is invoked.
- [SetOutOfProcessRecogTimeout](#)
Sets the number of seconds to wait before it is determined that a recognition action is no longer running properly.
- [SetRecogFailureRetryDelay](#)
Sets the number of seconds to wait before restarting a failed recognition action.

- [SetupAutomaticRetry](#)
Changes the settings that govern the automatic retry mechanism.
- [SnapCCOtoDCO](#)
Transfers the recognition results in the current page's CCO file - its Fingerprint file - to the appropriate Field objects of the Document Hierarchy...its setup DCO.
- [SnapDCOtoCCO](#)
Transfers the recognition results assigned to Field objects of the Document Hierarchy (aka the setup DCO) to the current page's CCO file, also known as its Fingerprint file.
- [SnapFieldtoChars](#)
Adjusts the zone position of the passed dco field to the field's character positions.
- [UseOutOfProcessRecog](#)
Causes recognition to be performed in a process that is separate from the process that is running the recognition actions.

Parent topic: [Global actions](#)

AnalyzeImage

Converts the Image file (.tif) that represents the current page to a Fingerprint file (.cco) file for the page.

Member of namespace

Recog_Shared

Syntax

```
bool AnalyzeImage ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Page object of the Document Hierarchy. Otherwise, True.

Level

Page level.

Details

This action creates the following items:

CCO fingerprint file	Converts the Image file (.tif) that represents the current page to a CCO fingerprint file for the page. The fingerprint file is a condensed representation of the dark areas on the page. These files can be used with the FindFingerprint and related actions, but they are not compatible with fingerprints that are created by OCR actions. Also, they do not contain text that can be used with locate actions or for click-n-key data entry.
CCOFILE variable	Creates a CCOFILE variable on the current page to indicate the successful creation of the CCO file.

A ruleset with this action should be bound to a Page object that represents an application's source page.

The action is not required if full-page recognition takes place using actions such as `RecognizePageOCR_S` or `RecognizePageICR_C`.

Important: Fingerprint matching accuracy can decline for images with a very large number of small dots. This condition might be due to large dotted or shaded areas on the original document, speckled noise in the scanned image, or images with text characters that are very broken up. If the fingerprint matching is too inaccurate for your purposes, use full-page recognition instead of this action. For more information, see [Changing the fingerprint creation method](#).

Example

```
AnalyzeImage ()
SetProblemValue (0.5)
SetSearchArea (0.5)
FindFingerprint (True)
```

This sequence generates a CCO file for the current page and attempts to match the current page with a fingerprint. For more information about the matching process, see the descriptions of the AutoDoc actions.

Parent topic: [Recog_Shared actions](#)

CCONormalization_OFF

Prevents the automatic running of NormalizeCCO procedures after a full-page recognition action was run.

Member of namespace

Recog_Shared

Syntax

```
bool CCONormalization_OFF ()
```

Parameters

None.

Returns

False if the action does not run at the Page level. Otherwise, True.

Level

Page level.

Details

A full-page recognition action such as `RecognizePageICR_C` automatically calls the `NormalizeCCO` action, which is thorough but time-consuming, after recognition is complete. This action is part of the `cco2cco.rrx` file.

To bypass this procedure, place `CCONormalization_OFF` just before the recognition action.

Example

```
CCONormalization_OFF()  
RecognizePageICR_C()
```

Parent topic: [Recog_Shared actions](#)

CreateTextFile

Creates a Text file (.txt) for the current page; adds the page's recognized values to the file; and places the file in the current batch, in your application's Batches directory.

Member of namespace

Recog_Shared

Syntax

```
bool CreateTextFile ()
```

Parameters

None.

Returns

False if the ruleset with this action is not bound to a Page object of the Document Hierarchy, or if an Image file for the current page is not available. Otherwise, True.

Level

Page only.

Details

This action creates a UTF-8 encoded Text file (.txt) for the current page; adds the page's recognized values to the file; and places the file in the current batch, in your application's Batches directory.

Attention: The Text file generated by this action is handy for debugging purposes, to see what recognition is placing into the page's Fingerprint file (.cco) file. The action should follow a full-page recognition action such as RecognizePageOCR_S, in a rule that is applied to a Page object of the Document Hierarchy.

Example

```
RecognizePageOCR_S()  
CreateTextFile()  
SetProblemValue(0.7)  
SetSearchArea(0.5)  
FindFingerprint(True)
```

After the full-page recognition action (RecognizePageOCR_S), the CreateTextFile() action places the recognized values into a Text file that it has set up for the page, and adds the file to the current batch, in the Batches directory of the application.

The text file that is created has the same filename as the image, but is assigned a .txt filename extension.

Parent topic: [Recog_Shared actions](#)

IsBlankPage

Counts the number of words in the Fingerprint file (.cco) file of the current page and returns True if the count is less than or equal to the number you enter as the parameter.

Member of namespace

Recog_Shared

Syntax

```
bool IsBlankPage (StrParam)
```

Parameters

Long value indicating the maximum number of words in the Fingerprint file (.cco) of a blank source page. "50", for example, tells the action that if a CCO file has 50 words or less, its page is blank. Valid values are 0 to 2,147,483,647.

Returns

False if the action parameter is invalid or if the action is unable to locate the Image file for the current page or its CCO file. Otherwise, True.

Level

Page level.

Details

This action counts the number of words in the CCO file of the current page and returns True if the count is less than or equal to the number you enter as the parameter.

A rule containing this action should apply to a Page object; within the rule, this action should come after one of the actions that creates a fingerprint, such as AnalyzeImage, RecognizePageOCR_S, or RecognizePageICR_C.

Example

```
AnalyzeImage ()  
IsBlankPage (5)  
SetPageType (Separator)
```

This sequence uses AnalyzeImage to create a CCO file, then checks to see if the file contains less than six words. If so, the IsBlankPage(5) action returns True. The final action, a DCO action, establishes the page as a Separator page.

Parent topic: [Recog_Shared actions](#)

RecogContinueOnFailure

Determines if a batch will abort if page or field recognition fails.

Important: The intended use of this legacy action is limited to those action libraries that do not support automatic retry. For all other libraries, the continue-on-failure setting is automatically set to True, which means that batches are not aborted as a result of recognition action failures. For information about automatic retry and the action libraries that support it, see [SetupAutomaticRetry](#).

Member of namespace

Recog_Shared

Syntax

```
bool RecogContinueOnFailure (StrParam)
```

Parameters

String value True or False.

1. True: a recognition failure will not be automatically retried if recognition fails. The batch will continue and the application can use the value assigned to the *RecogStatus* variable to decide how to proceed on success or failure of recognition. For more information on the *RecogStatus* variable, see information about the *RecogContinueOnFailure* action.
2. False: causes the batch to abort if a full-page or field-level recognition action fails. If *UseOutOfProcessRecog* is enabled, the batch will abort only if the second recognition attempt fails.

Returns

Always True.

Level

All.

Details

This action determines if a batch will abort if page or field recognition failed.

Note: If *RecogContinueOnFailure* is not specifically called, the default False value is used. This means that batches will abort if recognition fails.

After a recognition operation is complete, the variable *RecogStatus* is set to indicate the success or failure of recognition. If page-level recognition is being performed, *RecogStatus* values of 0, 1 or 2 are considered successful.

The full list of values includes:

- 0 - Success
- 1 - Recognition was successful but there are no results, the page was empty.
- 2 - Recognition was successful and additional processing such as *RotateImage* was performed.
- 4 - Failure: the recognition engine cannot be instantiated.
- 5 - Failure: the recognition engine timed out (the time specified by the *SetEngineTimeout* action has expired).
- 6 - Failure: could not load image to engine.
- 7 - Failure: could not load image to engine (path not found).
- 8 - Failure: image could not be rotated.
- 10 - Failure: general failure occurred and recognition was not completed.

Example

```
RecogContinueOnFailure(True)  
RecognizePageOCR_S()
```

Parent topic: [Recog_Shared actions](#)

RecogOMRThreshold

Performs OMR check box recognition by counting black pixels within each OMR box area in a Field with one or more OMR boxes.

Member of namespace

Recog_Shared

Syntax

```
bool RecogOMRThreshold (string threshold, string background)
```

Parameters

String: threshold

String: background

Parameters

Floating or integer values that specify the count of black pixels in OMR boxes:

1. Threshold: the percentage of pixels in the zone. The field zone that is not the printed box that must be considered checked. For example, the lightest box that is not just noise, but must be considered a check mark.
2. Background: the percentage of pixels in the zone that might be due to scanner noise or the border of the printed box. This value also controls the range on either side of the Threshold value that is low confidence.

The action also accepts parameters that are fractional percentages, which are needed to detect marks in large zones.

The parameters must be experimentally adjusted on real-world scanned forms. First, determine the Threshold value that correctly identifies a light mark as checked, and correctly identifies noisy zones as cleared. Second, adjust the value of the Background parameter to achieve an acceptable confidence interval.

Returns

Always True.

Level

Field level.

Details

This action performs OMR check box recognition by counting black pixels within each OMR box area in a Field with one or more OMR boxes.

- **Text Boxes:** The action sets the text value of the field to a string of 0's and 1's (one digit per OMR box). It assigns a Confidence String to the string of digits: 4 for Low Confidence up to 9 for High Confidence.
- **Density String and Confidence Value:** The action also establishes a *DensityString* variable for the Character String, indicating percentage-filled, from ASCII 48 ('0') through 148. For each possible OMR box, is a character. The ASCII value of the character minus 48 is the percentage-filled. If the Density String=0X, the first OMR field was blank, and the second was 40% filled. The ASCII value for X is 88. 88 minus 48 = 40.
- **MultiPunch and Confidence Values:** If the MultiPunch setting is set to 1 and multiple OMRs are filled beyond the threshold, the one that was filled the most is marked and set to Low Confidence.
 - If the percentage-filled is below the second parameter, the OMR box is not selected and the confidence is high.
 - If the percentage-filled is between the two parameters, the OMR box is not selected and the confidence is low.
 - If the percentage-filled is above the first parameter and below double the first parameter minus the second parameter, the OMR box is selected and the confidence is low.
 - If the percentage-filled is above double the first parameter minus the second parameter, the OMR box is selected and the confidence is high.

Note: The `RecogOMRThreshold` action works best on dropout boxes, but with an appropriate background value can work effectively with boxes that are visible in the scanned image.

If you are using small visible boxes on your image, it is best to zone the area by surrounding the entire visible box with room for alignment movement. Then, factor out the black from the box by using the parameters. If you attempt to zone inside the borders of a visible box, you can get a false positive if the page does not align exactly.

The page image must be a 1 bit black and white image. If the image is not 1 bit, the action `C2BW_Convert` in the `ColorToBW` library can be used to convert the image to a black and white 1 bit image.

Examples

For a small to medium size zone, 10% filled might be considered a deliberate mark. Anything below 5% (Background) is not a mark. Anything above 15% ($\text{Threshold} + (\text{Threshold} - \text{Background})$) is a high confidence mark. This works with a non-dropout OMR field where the printed outline of the box takes up less than 5% of the zone area. It also works for dropout forms.

```
RecogOMRThreshold("10", "5")
```

The following example is for a signature line, or a large zone where the percentage-filled is much lower than for a small zone. This example assumes there is low background or noise.

```
RecogOMRThreshold("2", "0")
```

Parent topic: [Recog_Shared actions](#)

RegisterPageFields

Returns the field positions for all zoned fields of the current page.

Member of namespace

Recog_Shared

Syntax

```
RegisterPageFields ()
```

Parameters

None.

Returns

False if a ruleset with this action is not bound to a Page object of the Document Hierarchy; or if the action cannot find the page's Fingerprint file (.cco). Otherwise, True.

Level

Page level.

Details

This action returns the field positions of all zoned fields of the current page. The action is similar to the ReadZones action of the Zones.rrx file.

Note: Use the ReadZones action when possible.

Example

```
RegisterPageFields ()
```

Parent topic: [Recog_Shared actions](#)

RotateTio

Checks if an Image file processed by the ImageFix action that assigns a .tio extension to the file needs to be rotated by 90, 180, or 270 degrees. If so, the action rotates and then saves the Image file with the same .tio extension.

Member of namespace

Recog_Shared

Syntax

```
bool RotateTio (StrParam)
```

Parameters

A String value:

- True to initiate rotation
- False to prevent rotation

Returns

Always True.

Level

Page level.

Details

This action checks if an Image file processed by the ImageFix action that assigns the .tio extension needs to be rotated by 90, 180, or 270 degrees. If rotation is necessary, the action saves the Image file with the same .tio extension.

Example

```
AnalyzeImage ()
RotateTio (True)
RotateImage ()
RecognizePageICR_C ()
```

Parent topic: [Recog_Shared actions](#)

SetAdjustFieldToChars

Optional setting for SnapCCOtoDCO to adjust the field position to its character positions.

Member of namespace

Recog_Shared

Syntax

```
bool SetAdjustFieldToChars (StrParam)
```

Parameters

A String value:

- True to snap character positions
- False to disable snapping

Returns

True.

Level

Page or Field level.

Details

This action has SnapCCOtoDCO adjust the field position (parameter True) to the character positions results after snapping the character values to the field. Off by Default

Example

```
SetAdjustFieldToChars (TRUE)
SnapCCOtoDCO ()
```

Parent topic: [Recog_Shared actions](#)

SetFingerprintRecogPriority

Sets the option that controls whether a full-page recognition action is to create a Fingerprint file(.cco) - aka a CCO file - for the current page.

Member of namespace

Recog_Shared

Syntax

```
bool SetFingerprintRecogPriority (StrParam)
```

Parameters

String value: True or False to control the creation of the CCO.

- True: If a CCO already exists prior to recognition, it is replaced with a brand new one with recognition results.
- False: If SetFingerprintRecogPriority is not called or is set to False and a CCO already exists prior to recognition, the recognition results will be added to that CCO.

Returns

Always True.

Level

All.

Details

This action sets the option that controls whether a full-page recognition action is to create a CCO file for the current page. When the option is On, processing is faster because the call to the AnalyzeImage action is eliminated.

The difference between creating a CCO from scratch with recognition results and adding the recognition results to the existing CCO created by AnalyzeImage is that in the adding case, the recognized characters are put into the CCO in a manner that uses a different fingerprinting technique.

Note: Be sure to place this action before a full-page recognition action.

Example

```
SetFingerprintRecogPriority (True)
```

Parent topic: [Recog_Shared actions](#)

SetFullPageRecogArea

An optional action that sets the area of the current page that is to be the target of recognition procedures, when full-page recognition action is invoked.

Member of namespace

Recog_Shared

Syntax

```
bool SetFullPageRecogArea (StrParam)
```

Parameters

A decimal value indicating the percent of the page to be recognized in response to this action.

For example: "0.1" designates the first 10% of the page, while "1.0" calls for recognition of the entire page.

This action is helpful if you know that a page's values will always be in a particular location on the page, but recognition of the entire page is not necessary.

Returns

False if the ruleset with this action is not bound to a Page object of the Document Hierarchy, or if the action's parameter is not a decimal value. Otherwise, True.

Level

Page level.

Details

This optional action sets the area of the current page that will be the target of recognition procedures when full-page recognition action is called. For example: "0.1" indicates that the first 10% of the page is to be recognized; "1.00" indicates that the entire page is to be recognized.

Example

```
SetFullPageRecogArea (0.5)
```

Parent topic: [Recog_Shared actions](#)

SetOutOfProcessRecogTimeout

Sets the number of seconds to wait before it is determined that a recognition action is no longer running properly.

Important: The intended use of this legacy action is limited to those action libraries that do not support automatic retry. For all other libraries, the timeout setting for out-of-process recognition is automatically configured, which means that this action does not need to be called. For information about automatic retry and the action libraries that support it, see [SetupAutomaticRetry](#).

Member of namespace

Recog_Shared

Syntax

```
bool SetOutOfProcessRecogTimeout (StrParam)
```

Parameters

Numeric value that indicates the number of seconds to wait to determine that a recognition action is stalled or exited.

Returns

Always True.

Level

All.

Details

This action sets the number of seconds to wait before it is assumed that a recognition action is no longer running correctly. When the timeout is reached, the recognition process is removed from memory. The `SetOutOfProcessRecogTimeout` action is effective only when out-of-process recognition is enabled by the use of a `UseOutOfProcessRecog` action.

If a recognition action does not complete within the specified number of seconds indicated by a `SetOutOfProcessRecogTimeout` action or a `SetEngineTimeout` action, it is assumed that the recognition engine encountered a severe error. It is removed from memory and recognition automatically restarts one more time. If the recognition action completes successfully within the specified time on either the first or second attempt, that recognition action is successful. If the recognition action does not complete by the specified time on the second attempt, the recognition action is set to abort, if `RecogContinueOnFailure(False)` was used.

If `SetOutOfProcessRecogTimeout` is not called, the default value of 300 seconds is used. In normal conditions, the default value is sufficient and does not need to be changed. This value needs to be increased only if a single page consistently takes more than 5 minutes to complete, which is not a typical situation. The programmer can choose to shorten this time to reduce the time to detect failures earlier, provided there is time to perform recognition in worst case scenarios. For best results, you can set the timeout to be the same or longer than the value specified in a `SetEngineTimeout` action.

When a `SetOutOfProcessRecogTimeout` action is called, the setting is in effect for the entire batch so that you can set the value once, then call as many recognition actions as you want.

Example

```
SetOutOfProcessRecogTimeout (300)
UseOutOfProcessRecog (True)
RecognizePageOCR_S()
```

Parent topic: [Recog_Shared actions](#)

SetRecogFailureRetryDelay

Sets the number of seconds to wait before restarting a failed recognition action.

Member of namespace

Syntax

```
bool SetRecogFailureRetryDelay (StrParam)
```

Parameters

Numeric value indicating the number of seconds to wait before restarting a failed recognition action, and automatically reactivating recognition one more time.

Returns

Always True.

Level

All.

Details

This action sets the number of seconds to wait after the time specified in either a `SetOutOfProcessRecogTimeout` action or a `SetEngineTimeout` action has expired. Once either timeout has occurred, the recognition engine is removed from memory: the action will then wait the additional time specified by the `SetRecogFailureRetryDelay` action to be sure that the engine has exited before restarting recognition. `SetRecogFailureRetryDelay` only has an effect if out-of-process recognition has been enabled by a `UseOutOfProcessRecog` action.

If a recognition action does not complete within the number of seconds specified by a `SetOutOfProcessRecog` action or a `SetEngineTimeout` action, it is assumed that the recognition engine has encountered a severe error and that recognition will automatically be restarted one more time. If the recognition action completes successfully within the specified time on either the first or second attempt, that recognition action will be successful. If the recognition action does not complete by the specified time on the second attempt, the recognition action will be set to abort if `RecogContinueOnFailure(False)` has been used.

If `SetRecogFailureRetryDelay` is not specifically called, the default value of 10 seconds is used. Under normal conditions, the default value will be sufficient and does not need to be changed. This value needs to be increased only if a log indicates that errors are occurring when attempting to restart a failed recognition action, and the problem can be diagnosed by setting the `RecogStatus` to "4".

When `SetRecogFailureRetryDelay` is called, its setting will be in effect for the entire batch. This allows you to set the value once, and call as many recognition actions as necessary.

Example

```
SetRecogFailureRetryDelay(10)
UseOutOfProcessRecog(True)
RecognizePageOCR_S()
```

Parent topic: [Recog_Shared actions](#)

SetupAutomaticRetry

Changes the settings that govern the automatic retry mechanism.

Important:

- Automatic retry mode is enabled by default. For information about the action libraries that support automatic retry, see the Details section of this action.
- If automatic retry mode is not currently enabled, calling this action enables automatic retry mode. For information about legacy mode, see [SetEngineTimeoutOCR_S](#).

Member of namespace

Recog_Shared

Syntax

```
bool SetupAutomaticRetry (string retryCount, string timeout)
```

Parameters

Smart parameters are supported.

retryCount

The number of times that a failed action is to be retried. If you do not call this action, the default retryCount is 1.

timeout

The number of seconds to wait for an action to complete before the action is retried. (This number is the minimum wait time. Depending on the specific nature of the failure, the operation might not be retried until several seconds after the timeout expires.)

If this action is not called, the default timeout is 180 seconds. This timeout is long enough for most situations, but you might want to change it in some cases. For example, your application might process atypical documents that take longer than normal to complete successfully.

Returns

Always True.

Level

Any level. In a hierarchical way, the retry settings take effect for all DCO objects at the calling level and below the calling level. This level is the level at which you call this action. Here are some examples of the way that the hierarchy works:

Calling level	Objects affected
Batch	All objects in the batch, which includes objects at the document, page, and field levels.
Page	The page object and all fields that are attached to that page.
Batch and page	Both the fields on the page and the page itself use the page level setting while other objects use the batch level setting.

Details

For actions and action libraries that support automatic retry, use this action to change the settings that govern the retry logic. Currently, the OCR/SR action library supports automatic retry.

The intent of automatic retry is to provide transparent recovery for the rare situation in which an operation does not terminate. For example, a peculiar or invalid image might cause the recognition engine to never complete the recognition step. The step can sometimes complete successfully when it is retried.

The logic of the automatic retry mechanism consists of the following elements:

Completion timeout	The mechanism waits for the specified number of seconds for an action to complete.
Timeout expiration	If the action does not complete, the current operation is stopped and retried.
Retry success	If the action is successful after the automatic retry, the action completes and the operation continues as normal.
Retry failure	If the action is not successful, failure is indicated in the usual manner. For details, see the help for the action that failed.

Example:

In the following example, the default value for the timeout is changed right before the attempt to recognize the page.

```
setFingerprinntRecogPriority("TRUE")
SetupAutomaticRetry(1, 240)
RecognizePageOCR_S()
CreateTextFile()
```

Parent topic: [Recog_Shared actions](#)

Related reference:

[OCR_SR actions](#)

SnapCCOtoDCO

Transfers the recognition results in the current page's CCO file - its Fingerprint file - to the appropriate Field objects of the Document Hierarchy...its setup DCO.

Member of namespace

Recog_Shared

Syntax

```
bool SnapCCOtoDCO ()
```

Parameters

None.

Returns

False if a ruleset with this action is not bound to a Page object or Field object of the Document Hierarchy. Otherwise, True.

Level

Page or Field level.

Details

This action transfers the recognition results of the current page's CCO file to the appropriate Field objects of the Document Hierarchy (DCO). Note that the action only transfers values to Field objects. SnapCCOtoDCO will only clear / update field text when all of the following conditions are met:

- Field is not an OMR field, for example var *RecogType*=4).
- Field has positions assigned.
- Field does not have the variable *v_skipsnap* set to 1.
- Field has data mapping to the CCO, at least one character.
- Fixes issue that would affect processing of reserved fields. For example, fields that are used for anchor finding are followed by snapping of data.

Example

```
SnapCCOtoDCO ()
```

Parent topic: [Recog_Shared actions](#)

SnapDCOtoCCO

Transfers the recognition results assigned to Field objects of the Document Hierarchy (aka the setup DCO) to the current page's CCO file, also known as its Fingerprint file.

Member of namespace

Recog_Shared

Syntax

```
bool SnapDCOtoCCO ()
```

Parameters

None.

Returns

False if a rule with this action is not applied to a page. Otherwise, True.

Level

Page level.

Details

This action transfers the recognition results assigned to Field objects of the Document Hierarchy (DCO) to the current page's CCO file.

If zonal recognition is used instead of full-page recognition, the action will populate the current page's CCO file with the results of zonal recognition. Then, when the Verify task runs, a user can use the ClickNKey option to populate fields.

Example

SnapDCOtoCCO ()

Parent topic: [Recog_Shared actions](#)

SnapFieldtoChars

Adjusts the zone position of the passed dco field to the field's character positions.

Member of namespace

Recog_Shared

Syntax

```
bool SnapFieldtoChars (string Smartparam)
```

Parameters

String: Smartparam

Parameters

A Smart Parameter value representing a valid Field location.

Returns

False if a valid DCO field is not returned from the Smart parameter value. Otherwise, True.

Level

Any level.

Details

This action adjusts the field position of the passed DCO to the DCO's character positions. If the field does not have a text value, no adjustment to the field zone is performed.

Example

```
SnapFieldtoChars (@F)
```

Parent topic: [Recog_Shared actions](#)

UseOutOfProcessRecog

Causes recognition to be performed in a process that is separate from the process that is running the recognition actions.

Important: The intended use of this legacy action is limited to those action libraries that do not support automatic retry. For all other libraries, the out-of-process engine is automatically restarted. For information about automatic retry and the action libraries that support it, see [SetupAutomaticRetry](#).

Member of namespace

Syntax

```
bool UseOutOfProcessRecog (strParam)
```

Parameters

True: Recognition actions should run in a separate process.

False: Recognition should run in the same process as the recognition actions.

Returns

Always True.

Level

All.

Details

This action determines in which process recognition will be performed. Using a separate process for recognition provides an additional stability and automatic recovery ability as it will automatically retry a recognition action that runs into trouble, such as recognition that has stalled or unexpectedly terminated. The action must be placed before a full-page or field-level recognition action such as `RecognizePageOCR_S`.

The action is also directly tied to the `SetRecogFailureRetryDelay` action, which determines how long (in seconds) the `UseOutOfProcessRecog` action waits to determine that recognition has stopped responding and must be retried.

If the `UseOutOfProcessRecog` action is not specifically called, its default True setting will be used. If the action is called specifically, the True or False setting will be in effect for the entire batch. This allows you to set the value once, and call as many recognition actions as necessary.

Example

```
UseOutOfProcessRecog (True)
SetRecogFailureRetryDelay (10)
RecognizePageOCR_S ()
```

Parent topic: [Recog_Shared actions](#)

runner actions

Use the runner actions to do miscellaneous utility functions.

The runner actions can check batch integrity, manipulate the values of fields and variables, raise condition flags, and control rule execution.

- [AbortOnError](#)
Determines whether a task that encounters an error stops or continues.
- [CheckAllIntegrity](#)
Checks all documents in the batch to determine whether they meet the document integrity requirements that are specified in the document hierarchy (setup DCO).

- [CheckDocCount](#)
Determines whether the number of documents in the runtime hierarchy matches the expected document count as specified by the scan operator.
- [CheckPageCount](#)
Determines whether the number of pages in the runtime hierarchy matches the expected page count as specified by the scan operator.
- [DebugMode_OFF](#)
Disables enhanced logging.
- [DebugMode_ON](#)
Enables enhanced logging (disabled by default).
- [GoToNextFunction](#)
Returns False, which causes the next function in the ruleset to run.
- [MessageID](#)
Adds a runtime MESSAGE and a MessageID variable to the bound object of the document hierarchy.
- [MessageIDParameter](#)
Adds a runtime Value, Type, and Substitution index for use with a preconfigured MESSAGE and MessageID variable to the bound object of the Document Hierarchy.
- [PilotMessage_Clear](#)
Removes the MESSAGE variable from the current object.
- [PilotMessage_Set](#)
Assigns a message to the MESSAGE variable of the current object.
- [ProcessChildren](#)
Initiates the processing of elements that are represented by the bound object and its children.
- [rr_AbortBatch](#)
Stops processing the current batch and sets its status to Abort.
- [rr_Get](#)
Assigns the value of the specified variable to the Text property of the current object.
- [rr_WriteNode](#)
Creates a separate XML data file for the current object.
- [rrAppend](#)
Appends the value of the source object to the specified field.
- [rrCompare](#)
Compares the values of two variables and returns True if they are the same.
- [rrCompareCase](#)
Runs a comparison of two strings or smart parameters to see whether they are identical.
- [rrCompareCaseLength](#)
Uses the smart parameters that you enter as the parameter to locate and compare the values of two object's variables.
- [rrCompareNot](#)
Compares the values of two variables and returns False if they are the same.
- [rrCompareNotCase](#)
Negates the running of the rrCompareCase action. You can run this action in instances when two of the string or smart parameter values are different.
- [rrCompareNotCaseLength](#)
Negates the running of the rrCompareCaseLength action. You can run this action in instances when two of the values are different.
- [rrCompareNumeric](#)
Uses the Smart Parameters that you enter as the parameter to locate and compare the numeric values of two object's variables.
- [rrContains](#)
Uses the Smart Parameters that you enter as the parameter to locate and compare the values of two objects' variables.

- [rrCopy](#)
Copies the value, confidence levels, and positions from one field to another.
- [rrPrepend](#)
Inserts a value at the beginning of the specified field.
- [rrSet](#)
Assigns a value to a variable or field.
- [SetBatchPriority](#)
Sets the priority of the batch at the completion of the task.
- [SetOperatorID](#)
Sets the ID of the person who is operating Rulerunner.
- [SetReturnValue](#)
Returns True or False depending on the parameter that is specified.
- [SetStationID](#)
Sets the ID of the station where the person is operating Rulerunner.
- [SetTaskStatus](#)
Specifies the task status that is returned to an application as Abort, Canceled, Finished, Hold, or Pending when the current task completes.
- [SkipChildren](#)
Prevents the running of rules on child objects of the current object.
- [Status_Preserve_OFF](#)
Allows rules to change the STATUS value of fields, for example, to assign a problem status.
- [Status_Preserve_ON](#)
Prevents rules from changing the STATUS value of fields.
- [Task_NumberOfSplits](#)
Specifies the number of jobs the batch is sent to when a condition is raised before it returns to the main workflow.
- [Task_RaiseCondition](#)
Specifies the group index and the index of the condition to raise from the list on the Datacap Web Client Workflow tab. 0 is the first condition.

Parent topic: [Global actions](#)

AbortOnError

Determines whether a task that encounters an error stops or continues.

Syntax

```
bool AbortOnError (StrParam)
```

Parameters

True: Abort the batch if an error occurs.

False: Do not abort the batch if an error occurs.

Returns

False if the parameter is not True or False. Otherwise, True.

Level

All.

Details

Determines if tasks that encounter errors are to abort, or continue processing.

Example

```
AbortOnError("Yes")
```

Parent topic: [runner actions](#)

CheckAllIntegrity

Checks all documents in the batch to determine whether they meet the document integrity requirements that are specified in the document hierarchy (setup DCO).

Syntax

```
bool CheckAllIntegrity ()
```

Parameters

None.

Returns

True if the Document Integrity of the current batch meets the requirements as defined in the setup of the Document Hierarchy. Otherwise, False.

Level

Batch level.

Details

Checks that the documents in the batch contain the correct type and number of pages, in line with the Document Integrity requirements of the Document Hierarchy.

Example

```
CreateDocuments ()  
CheckAllIntegrity ()
```

These actions are part of a rule applied to the Batch object of the Document Hierarchy. The first assembles documents from the pages in the batch; the second ensures that the makeup of each document is valid.

Parent topic: [runner actions](#)

CheckDocCount

Determines whether the number of documents in the runtime hierarchy matches the expected document count as specified by the scan operator.

Syntax

```
bool CheckDocCount ()
```

Parameters

None.

Returns

True if the actual count is the same as the expected count. Otherwise, False.

Level

Batch level.

Details

The number of expected documents is usually provided by the operator of a job's Scan task. This very handy action can compare the actual amount to the estimate at any time after a CreateDocuments action has assembled the documents in the batch.

Example

```
CheckDocCount ()
```

Parent topic: [runner actions](#)

CheckPageCount

Determines whether the number of pages in the runtime hierarchy matches the expected page count as specified by the scan operator.

Syntax

```
bool CheckPageCount ()
```

Parameters

None.

Returns

True if the two counts are equal. Otherwise, False.

Level

Batch level.

Details

This action confirms that the number of actual images (pages) in the current task's Page file (.xml) matches the count of expected pages.

Example

```
CheckPageCount ()
```

Parent topic: [runner actions](#)

DebugMode_OFF

Disables enhanced logging.

Syntax

```
bool DebugMode_OFF ()
```

Parameters

None.

Returns

Always True.

Level

All.

Details

This actions turns off the enhanced logging procedures turned on by an earlier DebugMode_On action.

Enhanced logging expands the scope and depth of a processing log's information, and of the logs that a Rulerunner task generates when you are testing a rule and its actions.

This feature also increases a Log file's size significantly, and should only be used when you are testing the impact of an action and rule on the application's workflow.

Example

```
DebugMode_Off ()
```

Parent topic: [runner actions](#)

DebugMode_ON

Enables enhanced logging (disabled by default).

Syntax

```
bool DebugMode_ON ()
```

Parameters

None.

Returns

Always True.

Level

All.

Details

The following example shows enhanced logging during several actions.

Example

```
DebugMode_On ()
ExportOpenConnection (@APPVAR (values/dsn/exportdb:cs))
SetTableName (Invoice)
ExportFieldToColumn (Number, db_Number)
AddRecord ()
DebugMode_Off ()
```

Parent topic: [rrunner actions](#)

GoToNextFunction

Returns False, which causes the next function in the ruleset to run.

Syntax

```
bool GoToNextFunction ()
```

Parameters

None.

Returns

Always False.

Level

All.

Details

Returns a False condition so that the next function in the RuleSet can run.

Example

```
IsFieldMatching ("Skip")
GoToNextFunction ()
```

If the condition in the first action is met, the sequence assigns a False status to the second action and to the rule of which it is a part. As a result, execution continues with the next function in the Rule.

Parent topic: [runner actions](#)

MessageID

Adds a runtime MESSAGE and a MessageID variable to the bound object of the document hierarchy.

Syntax

```
bool MessageID (string Message, string Identifier)
```

Parameters

Two parameters:

1. The message with index substitution indicators to be assigned to the MESSAGE variable. Be sure to surround the message in quotation marks.
2. The ID to be assigned to the message to enable client-side translation.

Returns

Always True.

Level

All

Details

Adds a runtime MESSAGE and a MessageID variable to the bound object of the document hierarchy.

Example

```
MessageID("The field '{0}' has a value of '{1}'.", FieldValueMessage01)
```

Parent topic: [runner actions](#)

Related information:

[Application translation](#)

MessageIDParameter

»Adds a runtime Value, Type, and Substitution index for use with a preconfigured MESSAGE and MessageID variable to the bound object of the Document Hierarchy.«

Syntax

```
bool MessageIDParameter (string Value, string ValueType, int Index)
```

Parameters

Three parameters. Smart parameters are supported.

1. The runtime value that replaces a substitution indicator. Smart parameters are supported.

2. The value's `Type` to enable client-side translation lookup. Permitted `Type` values are `job`, `task`, `shortcut`, `field`, `workflow`, `appname`, `pagetype`, `doctype`, `text`, and `variable`.
3. The index of the substitution.

Returns

Always True.

Level

All

Details

»Adds a runtime Value, Type, and Substitution index for use with a preconfigured MESSAGE and MessageID variable to the bound object of the Document Hierarchy.«

Example

```
MessageID("The field '{0}' has a value of '{1}'.", FieldValueMessage01)
MessageIDParameter("@F.Name", field, 0)
MessageIDParameter("@F", Text, 1
```

Parent topic: [runner actions](#)

Related information:

[Application translation](#)

PilotMessage_Clear

Removes the MESSAGE variable from the current object.

Syntax

```
bool PilotMessage_Clear ()
```

Parameters

None.

Returns

Always True.

Level

All.

Details

Removes the runtime MESSAGE variable from the bound object of the Document Hierarchy.

Example

```
PilotMessage_Clear()
```

Parent topic: [rrunner actions](#)

PilotMessage_Set

Assigns a message to the MESSAGE variable of the current object.

Syntax

```
bool PilotMessage_Set (StrParam)
```

Parameters

The smart parameter message to be assigned to the *MESSAGE* variable. Be sure to surround the message in quotation marks.

Returns

Always True.

Level

All.

Details

Provides a runtime *MESSAGE* variable to the bound object of the Document Hierarchy, and assigns the Action's parameter as the variable value.

Example

```
PilotMessage_Set("Field +@F+ Value is not Valid")
```

Parent topic: [rrunner actions](#)

ProcessChildren

Initiates the processing of elements that are represented by the bound object and its children.

Syntax

```
bool ProcessChildren (StrParam)
```

Parameters

A two-part, comma-separated specification of a Condition and a Command.

The Condition is any valid VBScript expression. The Command is the VB executable that results from the Condition.

Returns

False if the number or sequence of the arguments are invalid. Otherwise, True.

Level

All.

Details

A follow-up action that initiates the processing of elements represented by the bound object, and all its children.

Example

```
ProcessChildren("1,Exit")
```

Parent topic: [rrunner actions](#)

rr_AbortBatch

Stops processing the current batch and sets its status to Abort.

Syntax

```
bool rr_AbortBatch ()
```

Parameters

None.

Returns

Always True.

Level

All.

Details

Stops processing the current batch and sets the status of the batch to Abort.

Example

```
rr_AbortBatch ()
```

Parent topic: [rrunner actions](#)

rr_Get

Assigns the value of the specified variable to the Text property of the current object.

Syntax

```
bool rr_Get (StrParam)
```

Parameters

A smart parameter referencing a value or which is a reference to a value that will be copied to the calling object.

Returns

False the parameter is missing. Otherwise, True.

Level

All.

Details

Uses the parameter's elements to locate the value of a source object's variable, and assign it to the calling object. If the calling object is a field, only the value of the field will be changed.

Example

```
rr_Get("@B.OPERATOR")
```

This example retrieves the value of the Batch object's Operator property and assigns it to the calling object's Text property, if the calling object is a field.

```
rr_Get("@DICT_WORD(..\MONTH)")
```

This example shows how Smart Parameters translates the OMR recognized value of the MONTH field to the text from a predefined dictionary. The text is then assigned to the calling object's *Text* property, if it is a field, or *Text* variable if it is not a field.

Parent topic: [rrunner actions](#)

Related reference:

[rrSet](#)

[rrCompare](#)

rr_WriteNode

Creates a separate XML data file for the current object.

Syntax

```
bool rr_WriteNode ()
```

Parameters

None.

Returns

Always True.

Level

All.

Details

Sets up a separate XML data file element for the calling object during Rulerunner processing.

Example

```
rr_WriteNode()
```

Parent topic: [runner actions](#)

rrAppend

Appends the value of the source object to the specified field.

Syntax

```
bool rrAppend (string varSource, string varTarget)
```

Parameters

Two Smart Parameters:

1. The source value.
2. A reference to the target field.

Both parameters are optional. If a parameter is not specified, it will default to the calling object. If the calling object is a field, it will use the field value.

Returns

False if the action cannot locate the target object or if the source value is empty. Otherwise, True.

Level

All

Details

The action retrieves the value of the source object, and appends it to the target value.

Example

```
rrAppend("@D.DocID", "@F")
```

This action inserts the current calling object's parent DocID variable value and appends it to the calling field's value.

Note: Target can not be a variable. If the source and target are the same, the action has no effect.

Parent topic: [runner actions](#)

Related reference:

[rrPrepend](#)

rrCompare

Compares the values of two variables and returns True if they are the same.

Syntax

```
bool rrCompare (string object1, string object2)
```

Parameters

Two Smart Parameters.

1. A value or a smart parameter, which is a reference to a value.
2. A value or a smart parameter, which is a reference to a value for comparison.

Note: Either reference can specify a variable of the calling object (the bound object of the Document Hierarchy.) Alternatively, both references can identify a variable of an object that is a parent or child of the calling object.

Either parameter is optional. If a parameter is not specified, it will default to the calling object. If the calling object is a field, it will use the field value. For batch, document and page objects, it will use a variable called *Text*, creating the variable if it does not exist.

Returns

False if the compared values do not match. Otherwise, True.

Level

All.

Details

Uses the Smart Parameters that you enter as the parameter to locate and compare the values of two object's variables.

Example

Code	Comment
<pre>rrCompare ("Expected_Pages", "@B.Tot_Pages")</pre>	Solicits a value from the field Expected_Pages of the calling object and compares it to the value of the field @B.Tot_Pages of the Batch object. The action returns False if the values are not the same.
<pre>rrCompare ("Expected Value", "@F\MyFieldToTest")</pre>	Checks the value of the field MyFieldToTest.
<pre>rrCompare ("MyExpectedName", "@TASKNAME")</pre>	Checks the current task name.
<pre>rrCompare ("@EMPTY", "@F\MyFieldToTest")</pre>	Checks whether the field MyFieldToTest contains an empty string.
<pre>rrCompare ("MyExpectedName", "@JOBNAME")</pre>	Checks the current job name.

Parent topic: [rrunner actions](#)

Related reference:

[rrCompareNot](#)

rrCompareCase

Runs a comparison of two strings or smart parameters to see whether they are identical.

Syntax

```
bool rrCompareCase(string object1, string object2, string caseSensitive)
```

Parameters

string object1

string object2

string caseSensitive

Parameters

Three Parameters.

1. A value or a smart parameter, which is a reference to a value.
2. A value or a smart parameter, which is a reference to a value for comparison.
Note: Either reference can specify a variable of the calling object (the bound object of the Document Hierarchy.) Alternatively, both references can identify a variable of an object that is a parent or child of the calling object.

Either parameter is optional. If a parameter is not specified, it defaults to the calling object. If the calling object is a field, it uses the field value. For batch, document and page objects, it uses a variable that is called *Text*, creating the variable if it does not exist.

3. True runs a case-sensitive compare. False runs a case insensitive compare. If not specified, the default is False.

Returns

False If the compared values do not match. Otherwise, True.

Level

All.

Details

Runs a comparison of two strings or smart parameters to see whether they are identical. The comparison can be run as case sensitive or case insensitive.

Example

```
rrCompareCase("Main_Job","JOBID","False")
```

This example compares the string "Main_Job" to the current Job ID. The comparison is case insensitive. If the current Job ID is "MAIN_JOB", the action returns True.

```
rrCompareCase("Main_Job","JOBID","True")
```

This example compares the string "Main_Job" to the current Job ID. The comparison is case-sensitive. If the current Job ID is "MAIN_JOB", the action returns False.

Parent topic: [rrunner actions](#)

Related reference:

[rrCompareNotCase](#)

[rrCompareNot](#)

[rrCompare](#)

rrCompareCaseLength

Uses the smart parameters that you enter as the parameter to locate and compare the values of two object's variables.

Syntax

```
bool rrCompareCaseLength(string object1, string object2,  
string caseSensitive, string length, string fromStart)
```

Parameters

string object1

string object2

string caseSensitive

string length

string fromStart

Parameters

Five Parameters.

1. A value or a smart parameter, which is a reference to a value.
2. A value or a smart parameter, which is a reference to a value for comparison.
3. True runs a case-sensitive compare. False runs a case insensitive compare.
4. An integer for the number of characters to compare. If the length is 0, the entire string is compared.
5. True compares from the start of the string. False compares from the end of the string.

Returns

False If the compared values do not match. Otherwise, True.

Level

All.

Details

Uses the Smart Parameters that you enter as the parameter to locate and compare the values of two object's variables. The comparison can be limited to a specified number of characters from the start or the end of the string. The comparison can be run case-sensitive or case insensitive.

Example

```
rrCompareCaseLength("Main", "@JOBID", "False", 4, "True")
```

This example compares the string "Main" to the current Job ID. Only the first four letters are compared and the compare is case-sensitive. If the current Job ID is "MAIN_JOB", the action returns True.

```
rrCompareCaseLength("Main", "@JOBID", "True", 4, "True")
```

This example compares the string "Main" to the current Job ID. Only the first four letters are compared and the compare is case-sensitive. If the current Job ID is "MAIN_JOB", the action returns False.

```
rrCompareCaseLength("Main Line", "Main Job", "True", 4, "True")
```

This example compares the string "Main Line" to the string "Main Job". The compare is case-sensitive and only the first four letters are compared. The action returns True.

```
rrCompareCaseLength("@P.ScanSrcPath", "GOOD.BMP", "False", 8, "False")
```

This example runs a case insensitive compare of the last 8 characters of the *ScanSrcPath* variable to find the last 8 characters of the string "GOOD.BMP". If the value of *ScanSrcPath* is "c:\test\testvalidate\images\good.bmp", the action returns True.

Parent topic: [runner actions](#)

Related reference:

[rrCompareNotCaseLength](#)

[rrCompareNot](#)

[rrCompare](#)

rrCompareNot

Compares the values of two variables and returns False if they are the same.

Syntax

```
bool rrCompareNot (string object1, string object2)
```

Parameters

Two Smart Parameters.

1. A value or a smart parameter, which is a reference to a value.
2. A value or a smart parameter, which is a reference to a value for comparison.

Note: Either reference can specify a variable of the calling object. Alternatively, both references can identify a variable of an object that is a parent or child of the calling object.

Either parameter is optional. If a parameter is not specified, it will default to the calling object. If the calling object is a field, it will use the field value.

For batch, document and page objects, it will use a variable called *Text*, creating the variable if it does not exist.

Returns

True if the compared values do not match. Otherwise, False.

Level

All.

Details

This action is the negation of `rrCompare`. It can be handy for when an action should be performed only when two values are different.

Example

```
rrCompareNot ("Expected_Pages", "@B.Tot_Pages")
rr_AbortBatch ()
```

This example shows how a value is solicited from the field `Expected_Pages` off of the calling object and the `Batch` object. The two values are then compared: the action returns `True` if the values are not the same. Here, the batch will abort if the expected pages do not match the total pages.

Parent topic: [rrunner actions](#)

rrCompareNotCase

Negates the running of the `rrCompareCase` action. You can run this action in instances when two of the string or smart parameter values are different.

Syntax

```
bool rrCompareNotCase (string object1, string object2, string caseSensitive)
```

Parameters

string object1

string object2

string caseSensitive

Parameters

Three Parameters.

1. A value or a smart parameter, which is a reference to a value.
2. A value or a smart parameter, which is a reference to a value for comparison.
Note: Either reference can specify a variable of the calling object (the bound object of the Document Hierarchy.) Alternatively, both references can identify a variable of an object that is a parent or child of the calling object.

Either parameter is optional. If a parameter is not specified, it defaults to the calling object. If the calling object is a field, it uses the field value. For batch, document and page objects, it uses a variable that is called *Text*, creating the variable if it does not exist.

3. `True` runs a case-sensitive compare. `False` runs a case insensitive compare. If not specified, the default is `False`.

Returns

True if the compared values do not match. Otherwise, False.

Level

All.

Details

This action negates the running of the `rrCompareCase` action. You can run this action in instances when two of the string or smart parameter values are different.

Example

```
rrCompareNotCase("Main_Job", "JOBID", "False")
```

This example compares the string "Main_Job" to the current Job ID. The comparison is case insensitive. If the current Job ID is "MAIN_JOB", the strings match so the action returns False.

```
rrCompareNotCase("Main_Job", "JOBID", "True")
```

This example compares the string "Main_Job" to the current Job ID. The comparison is case-sensitive. If the current Job ID is "MAIN_JOB", the strings do not match so the action returns True.

Parent topic: [runner actions](#)

Related reference:

[rrCompareCase](#)

[rrCompareNot](#)

[rrCompare](#)

rrCompareNotCaseLength

Negates the running of the `rrCompareCaseLength` action. You can run this action in instances when two of the values are different.

Syntax

```
bool rrCompareNotCaseLength(string object1, string object2,  
string caseSensitive, string length, string fromStart)
```

Parameters

string object1

string object2

string caseSensitive

string length

string fromStart

Parameters

Five Parameters.

1. A value or a smart parameter, which is a reference to a value.

2. A value or a smart parameter, which is a reference to a value for comparison.
3. True runs a case-sensitive compare. False runs a case insensitive compare.
4. An integer for the number of characters to compare. If the length is 0, the entire string is compared.
5. True compares from the start of the string. False compares from the end of the string.

Returns

True, if the compared values do not match. Otherwise, False.

Level

All.

Details

Uses the Smart Parameters that you enter as the parameter to locate and compare the values of two object's variables. The comparison can be limited to a specified number of characters from the start or the end of the string. The comparison can be run case-sensitive or case insensitive.

Example

```
rrCompareNotCaseLength("Main", "@JOBID", "False", 4, "True")
```

This example compares the string "Main" to the current Job ID. Only the first four letters are compared and the compare is case-sensitive. If the current Job ID is "MAIN_JOB", the action returns False.

```
rrCompareNotCaseLength("Main", "@JOBID", "True", 4, "True")
```

This example compares the string "Main" to the current Job ID. Only the first four letters are compared and the compare is case-sensitive. If the current Job ID is "MAIN_JOB", the comparison does not match due to case differences so the action returns True.

```
rrCompareNotCaseLength("Main Line", "Main Job", "True", 4, "True")
```

This example compares the string "Main Line" to the string "Main Job". The compare is case-sensitive and only the first four letters are compared. The comparison matches so the action returns False.

```
rrCompareNotCaseLength("@P.ScanSrcPath", "GOOD.BMP", "False", 8, "False")
```

This example runs a case insensitive compare of the last 8 characters of the *ScanSrcPath* variable to find the last 8 characters of the string "GOOD.BMP". If the value of *ScanSrcPath* is "c:\test\testvalidate\images\good.bmp", the comparison matches so the action returns False.

Parent topic: [runner actions](#)

Related reference:

[rrCompareCaseLength](#)

[rrCompareNot](#)

[rrCompare](#)

rrCompareNumeric

Uses the Smart Parameters that you enter as the parameter to locate and compare the numeric values of two object's variables.

Syntax

```
bool rrCompareNumeric( value1, operator, value2)
```

Parameters

Two Smart Parameters and one Operator.

1. A numeric value or a smart parameter, which is a reference to a numeric value.
2. Comparison operator as second parameter. Operator can be "<", ">", or "="
3. A numeric value or a smart parameter, which is a reference to a numeric value, for comparison.

Note: Either reference can specify a variable of the calling object (the bound object of the Document Hierarchy.) Alternatively, both references can identify a variable of an object that is a parent or child of the calling object.

Returns

False, if the parameters to be compared are not numeric, or if operator is not specified, or if the condition is not satisfied. Otherwise, returns True.

Level

All.

Details

Either of the two parameters to be compared is optional. If a parameter is not specified, the action defaults to the calling object. If the calling object is a field, the action uses the field value. For batch, document and page objects, the actions uses the variable *Text*, creating the variable if it does not exist.

Example

```
rrCompareNumeric("@P\Total_Cost", >, "100")
```

Assuming that the rule is configured at page level and getting the "Total_Cost" field value present in that page. If the value in field "Total_Cost" is greater than 100, then the action returns TRUE, else it returns FALSE.

```
rrCompareNumeric("@P.TemplateID", =, "555")
```

This example shows a value from a page variable *@P.TemplateID* to be compared with the value 555, which is the default fingerprint. If the *@P.TemplateID* is equal to 555, then the action returns TRUE, else it returns FALSE. From the result, it can be concluded whether the default fingerprint is being used.

Parent topic: [rrunner actions](#)

rrContains

Uses the Smart Parameters that you enter as the parameter to locate and compare the values of two objects' variables.

Syntax

```
bool rrContains(string object1, string object2, string caseSensitive)
```

Parameters

string object1
string object2
string caseSensitive

Parameters

Two Smart Parameters

1. A value or a smart parameter, which is a reference to a value.
2. A value or a smart parameter, which is a reference to a value, that can contain the value from argument 1.

Note:

Either reference can specify a variable of the calling object (the bound object of the Document Hierarchy). Alternatively, both references can identify a variable of an object that is a parent or child of the calling object.

Either parameter is optional. If a parameter is not specified, it defaults to the calling object. If the calling object is a field, it uses the field value. For batch, document, and page objects, it uses a variable named *Text*, creating the variable if it does not exist.

Returns

True, if the value of argument 1 is found in the value of argument 2. Otherwise, False.

Level

All.

Details

Uses the Smart Parameters that you enter as the parameter to locate and compare the values of two objects' variables.

Example

```
rrContains("Dr", "@P.Name")
```

This example shows how a value *Dr* is used as direct input, and the Page object's *Name* variable is solicited. The two values are then compared. The action returns False if the value *Dr* is not found in the value that is solicited from *@P.Name*.

Parent topic: [runner actions](#)

rrCopy

Copies the value, confidence levels, and positions from one field to another.

Syntax

```
bool rrCopy (string varSource, string varTarget)
```

Parameters

Two Smart Parameters

1. A reference to the source field
2. A reference to the target field

Either parameter is optional. If a parameter is not specified, the calling object must be a field.

Returns

False, if the action cannot retrieve the target or source object. Otherwise, True.

Level

Field level.

Details

The action retrieves the value, confidence, and image references (field positions) of the source field object, and copies them to the target field object. It uses the Smart Parameters that you enter as a parameter to copy the value of a source field object to a target field object. This action is unusual in that it is intended to work only on field objects.

Note: `rrCopy` copies more than just the value of the field. Use `rrSet` if only the field value is to be copied. This action is unusual because it is intended to work only on field objects.

Example

```
rrCopy("@B\OPERATOR", "@P\OPERATOR")
```

This example copies the Operator value of the Batch field to the Operator field of the bound object of the Document Hierarchy.

Parent topic: [runner actions](#)

Related reference:

[rrSet](#)

[rr_Get](#)

rrPrepend

Inserts a value at the beginning of the specified field.

Syntax

```
bool rrPrepend (string varSource, string varTarget)
```

Parameters

Two Smart Parameters:

1. The source value.
2. A reference to the target object.

Either parameter is optional. If a parameter is not specified, it will default to the calling object. If the calling object is a field, it will use the field value.

Returns

False if the calling object and target object are the same, if the action cannot locate the target object's variable, or if the source value argument or object returns an empty string. Otherwise, True.

Level

All, target must be a field object.

Details

The action retrieves the value of the source object, and pre-appends it to the target field value.

Example

```
rrPrepend("@D.DocID", "@F")
```

This action inserts the current calling object's parent DocID variable value and pre-appends it to the calling field's value.

Note: Target can not be a variable.

Parent topic: [rrunner actions](#)

Related reference:

[rrAppend](#)

rrSet

Assigns a value to a variable or field.

Syntax

```
bool rrSet (string varSource, string varTarget)
```

Parameters

Smart parameters are supported.

Both parameters for this action are optional. If you do not specify a parameter, the default item that the parameter references depends on the calling object type as shown in the following table:

Calling object type	Default item referenced
Field	The field that is the calling object
Page, Document, or Batch	The variable Text of the calling object. This variable is created if it does not exist.

varSource

The item whose value is to be copied to the target item.

varTarget

The item that is to receive the copied value.

Returns

False if the action cannot locate the target object. Otherwise, this action returns True.

Level

All.

Details

Use this action to copy the value of a source item to a target item.

To copy a field's value, confidence, and image references (field positions) of the source field object, use `rrCopy`.

Example

In these examples, the target item's value is set to be the same as the source item's value.

Code	Source	Target
<code>rrSet("@P\Date", "@F")</code>	The field Date that is a sibling of the target field.	Current field
<code>rrSet("@P.FieldCount", "@F")</code>	Page variable FieldCount	Current field
<code>rrSet("@F", "@P.Variable")</code>	Current field	Current page variable Variable
<code>rrSet("@F", "@F.Variable")</code>	Current field	Current field variable Variable
<code>rrSet("@P.TemplateID", "@P\F\nfieldName")</code>	Variable TemplateID	The field DCO object FieldName
<code>rrSet("@F.MySourceVar", "@P.\nMyTargetVar")</code>	The calling field MySourceVar variable	The MyTargetVar variable that belongs to the parent page of the calling object
<code>rrSet("@D.Tot_Pages", "@B.To\nt_Pages")</code>	The calling document's Tot_Pages variable. This example assumes that the calling object is a child of a Document object.	The Tot_Pages variable of the Batch object
<code>rrSet("@DICT_VALUE(..\nMONTH)", "")</code>	The OMR recognized value of the MONTH field as translated by the smart parameter to the text from a predefined dictionary	The calling object's Text property if the calling object is a field, or the Text variable if the calling object is not a field

Parent topic: [rrunner actions](#)

Related reference:

[rrCopy](#)

[rr_Get](#)

[rrCompare](#)

SetBatchPriority

Sets the priority of the batch at the completion of the task.

Syntax

```
bool SetBatchPriority (string Priority)
```

Parameters

A single value to update the batch priority at the end of the Task.

Returns

False if the value of the argument is invalid. Otherwise, True.

Level

All.

Details

Values are typically 1-9 with 5 being the median. Batches with priority 1 are processed first, batches with priority 9 are processed last.

Example

```
SetBatchPriority("1")
```

Parent topic: [runner actions](#)

SetOperatorID

Sets the ID of the person who is operating Rulerunner.

Syntax

```
bool SetOperatorID (string OperatorID)
```

Parameters

A Single value representing the new Operator ID value.

Returns

False if setting the value throws an error. Otherwise, True.

Level

All.

Details

Sets the Operator ID at the completion of the Task.

Example

```
SetOperatorID("admin")
```

Parent topic: [runner actions](#)

SetReturnValue

Returns True or False depending on the parameter that is specified.

Syntax

```
bool SetReturnValue (StrParam)
```

Parameters

True: The action will return true.

False: The action will return false.

Returns

True if the action is passed the parameter true. Otherwise, False.

Level

All.

Details

This action will return true or false based on the input parameter.

By passing in true, the action will return true and continue with the actions in the current function. If this is the last action in a function, any following functions within the rule are skipped.

One use for this action is a quick way to disable a rule by adding a new function, that precedes all other functions in the rule, where the new function contains only this action with a parameter of true. This will cause all other functions in the rule to be skipped and the next rule will run.

Using a parameter of false, this action will return false, causing all following actions in the function to be skipped and control carries forward to the next function in the same rule. In this way, the operation is identical to the action `GoToNextFunction`.

Example

```
SetReturnValue("true")
```

Parent topic: [runner actions](#)

Related reference:

[GoToNextFunction](#)

[FailRuleSet \(deprecated\)](#)

[rr_AbortBatch](#)

SetStationID

Sets the ID of the station where the person is operating Rulerunner.

Syntax

```
bool SetStationID (string StationID)
```

Parameters

A Single value representing the new Station ID value.

Returns

False if setting the value throws an error. Otherwise, True.

Level

All.

Details

Sets the Station ID at the completion of the Task.

Example

```
SetStationID("4")
```

Parent topic: [runner actions](#)

SetTaskStatus

Specifies the task status that is returned to an application as Abort, Canceled, Finished, Hold, or Pending when the current task completes.

Syntax

```
bool SetTaskStatus (StrParam)
```

Parameters

Numeric value representing the status that the task is to return to User Application. The statuses include:

- 0 - Abort
- 1 - Cancelled
- 2 - Finished
- 4 - Hold
- 8 - Pending

Returns

False if the parameter is not Numeric. Otherwise, True.

Level

All.

Details

Sets the Task Status value that is to be returned to User Application when the current task finishes processing.

Example

```
SetTaskStatus (4)
```

Parent topic: [runner actions](#)

SkipChildren

Prevents the running of rules on child objects of the current object.

Syntax

```
bool SkipChildren ()
```

Parameters

None.

Returns

Always True.

Level

All.

Details

Prevents rules applied to child objects of the current parent object from being run. The action can optimize the execution of rules by eliminating the need to visit every field on every page.

Example

```
SkipChildren ()
```

Parent topic: [runner actions](#)

Status_Preserve_OFF

Allows rules to change the STATUS value of fields, for example, to assign a problem status.

Syntax

```
bool Status_Preserve_OFF ()
```

Parameters

None.

Returns

Always True.

Level

All.

Details

This action turns the Status Preserve condition of a page and its fields from On to Off.

An object's Off condition allows the actions of a Validate ruleset to assign a problem status to any Field object with an invalid captured value. The Verify task's Data Entry panel will then surround the value with a pink background, alerting the operator to the problem.

Example

```
Status_Preserve_Off()
```

Parent topic: [runner actions](#)

Status_Preserve_ON

Prevents rules from changing the STATUS value of fields.

Syntax

```
bool Status_Preserve_ON ()
```

Parameters

None.

Returns

Always True.

Level

All.

Details

This action changes the Status Preserve condition of a Page object and its Field objects from Off to On.

The On condition prevents a rule and its actions from assigning a "problem" status to a field, even if the field's value fails validation.

Example

```
Status_Preserve_On()
```

Parent topic: [runner actions](#)

Task_NumberOfSplits

Specifies the number of jobs the batch is sent to when a condition is raised before it returns to the main workflow.

Syntax

```
bool Task_NumberOfSplits (nNumOfSplits)
```

Parameters

Integer value of the number of splits. In most cases, you will want to use "1" as the parameter.

Returns

False if the parameter you enter is not Numeric. Otherwise, True.

Level

All.

Details

Specifies how many times sub-batches have been created from the current batch.

Important: The action communicates but does not use the `Number_of_Splits` value you enter as a parameter.

Example

```
Task_NumberOfSplits(1)
Task_RaiseCondition(0,0)
```

In this example, the User Application is alerted to create one sub-batch entry, and to raise the second child job condition for this sub-batch entry.

Parent topic: [rrunner actions](#)

Related reference:

[Task_RaiseCondition](#)

Task_RaiseCondition

Specifies the group index and the index of the condition to raise from the list on the Datacap Web Client Workflow tab. 0 is the first condition.

Syntax

```
bool Task_RaiseCondition (strParam)
```

Parameters

Two comma-separated Integer values:

1. The applicable value of the sub-batch index. 0 is the first sub-batch, 1 is the second, etc. The `Task_NumberOfSplits` action determines how many sub-batches are created.
2. The value that designates the Child Job Condition that should be assigned to the specified sub-batch. 0 is the first Child Job Condition, 1 is the second, etc.

Returns

False if either parameter is not Numeric. Otherwise, True.

Level

All.

Details

Assigns the correct Child Job Condition to the correct sub-batch entry created by the Task_NumberOfSplits action.

Example

```
Task_NumberOfSplits(1)
Task_RaiseCondition(0,0)
```

In this example, the User Application is alerted to create one sub-batch entry, and to raise the second child job condition for this sub-batch entry.

Parent topic: [runner actions](#)

Related reference:

[Task_NumberOfSplits](#)

SharedRecognitionTools actions

Use the actions in this library after you perform OCR. For example, you might perform OCR with actions from the OCR_A or OCR_SR libraries and then use the actions in this library. Actions that can produce the layout XML include OCR_SR.Recognize and OCR_A.Recognize, both of which can process color images and PDF files. To use the Locate actions and perform click 'n' key during verification, use the action CreateCcoFromLayout to create a CCO file for the page after producing the layout XML file.

- [CreateCcoFromLayout](#)
Creates a CCO file after recognition or conversion is performed. The CCO is needed for the locate actions and for click 'n' key during verification.

Parent topic: [Global actions](#)

CreateCcoFromLayout

Creates a CCO file after recognition or conversion is performed. The CCO is needed for the locate actions and for click 'n' key during verification.

Syntax

```
bool CreateCcoFromLayout()
```

Parameters

None.

Returns

Always True.

Level

Page level.

Details

CreateCcoFromLayout requires a previously created layout file (for example: tm000001_layout.xml) where text is grouped into blocks. For more information about layout XML file, see [DocumentAnalytics actions](#).

Example

```
Recognize ()
CreateCcoFromLayout ()
```

Parent topic: [SharedRecognitionTools actions](#)

Related reference:

[Recognize](#)

Related information:

[Enabling automatic language detection for OCR_S recognition](#)

SignatureValidation actions

Use the Signature Validation actions to detect and validate signatures on a document or zone.

- [CreateNew](#)
Sets a value to determine if new signature records for signatures that do not match are to be added as reference signatures.
- [SetMinimumConfidence](#)
Sets the minimum confidence accepted for matched signatures.
- [SetSignatureReferenceFolderPath](#)
Sets the path to the folder where the signature reference files are stored.
- [ValidateSignature](#)
Performs signature detection and validation. This action should be called after the `CreateNew`, `SetMinimumConfidence`, and `SetSignatureReferenceFolderPath` actions.

Parent topic: [Global actions](#)

CreateNew

Sets a value to determine if new signature records for signatures that do not match are to be added as reference signatures.

Member of namespace

SignatureValidation

Syntax

```
bool CreateNew (string createNewSignatureReference , string
newReferenceSignatureFileName)
```

Parameters

createNewSignatureReference

True or False value indicating whether to add a signature that was not matched to the folder containing all the signature reference files. When set to true, it is expected that a signature that was not matched the first time, will be matched the second time it is processed. Smart parameters are supported.

`newReferenceSignatureFileName`

String value representing the name to use for the new signature file created when `createNewSignatureReference` is set to `True`.

Level

Page and field.

Returns

`True`, always.

Details

Sets a value to determine if new signature records for signatures that do not match are to be added as reference signatures.

For a typical application, it is recommended to have 5 to 10 references for a single signature. A person's signature can vary over time. It can also vary from day to day based on a number of factors such as mood, writing instrument, hand position, incline of the surface, etc. While the engine attempts account for these kind of differences, providing multiple examples of a reference signature helps to improve the accuracy of recognition.

```
SetSignatureReferenceFolderPath("@X.referenceFolderPath", "*" + @X.accountId + "*")
CreateNew("True", "@X.accountId")
SetMinimumConfidence("70")
ValidateSignature(False)
```

This example:

- sets the signature references directory folder path by using the `SetSignatureReferenceFolderPath` action, and compares only against the reference files with names that contain the value of a previously captured `accountId` DCO variable.
- instructs the signature matching algorithm to create a new reference file if the signature is not matched, and to name it using the value of a previously captured `accountId` DCO variable.
- sets the minimum confidence to 70 so that matching signatures with a confidence less than 70 are ignored.
- runs the signature detection/validation.

Parent topic: [SignatureValidation actions](#)

SetMinimumConfidence

Sets the minimum confidence accepted for matched signatures.

Member of namespace

SignatureValidation

Syntax

```
bool SetMinimumConfidence (string confidence)
```

Parameters

confidence

A value indicating the minimum confidence accepted for matched signatures. Valid values are in the range of 0-100. Smart parameters supported.

Level

Page and field.

Returns

True, always.

```
SetSignatureReferenceFolderPath("@X.referenceFolderPath", "*" + @X.accountId + "*")
CreateNew("True", "@X.accountId")
SetMinimumConfidence("70")
ValidateSignature(False)
```

This example:

- sets the signature references directory folder path by using the `SetSignatureReferenceFolderPath` action, and compares only against the reference files with names that contain the value of a previous captured `accountId` DCO variable.
- instructs the signature matching algorithm to create a new reference file if the signature is not matched, and to name it using the value of a previously captured `accountId` DCO variable.
- sets the minimum confidence to 70 so that matching signatures with a confidence less than 70 are ignored.
- runs the signature detection/validation.

Parent topic: [SignatureValidation actions](#)

SetSignatureReferenceFolderPath

Sets the path to the folder where the signature reference files are stored.

Member of namespace

SignatureValidation

Syntax

```
bool SetSignatureReferenceFolderPath (string referenceFileFolderPath , string
referenceFileFilter)
```

Parameters

referenceFileFolderPath

The path to the signature references directory. Smart parameters supported.

referenceFileFilter

Sets the filter to use when selecting reference files from the signatures reference directory. This is a DOS filter which supports the * and ? wild cards to determine which files from within the reference directory should be used to validate the signature. Smart parameters are supported.

Level

Page and field.

Returns

True, always.

Details

Sets the path to the folder where the signature reference files are stored. The directory can contain reference files for multiple signatures. Only the reference files that match the `referenceFileFilter` file mask will be used to match the target signature. For example, a filter of `*JohnSmith*` would match all reference files in the directory that have "JohnSmith" in the file name.

```
SetSignatureReferenceFolderPath("@X.referenceFolderPath", "*" + @X.accountId + *")
CreateNew("True", "@X.accountId")
SetMinimumConfidence("70")
ValidateSignature(False)
```

This example:

- sets the signature references directory folder path by using the `SetSignatureReferenceFolderPath` action, and compares only against the reference files with names that contain the value of a previously captured `accountId` DCO variable.
- instructs the signature matching algorithm to create a new reference file if the signature is not matched, and to name it using the value of a previously captured `accountId` DCO variable.
- sets the minimum confidence to 70 so that matching signatures with a confidence less than 70 are ignored.
- runs the signature detection/validation.

Parent topic: [SignatureValidation actions](#)

ValidateSignature

Performs signature detection and validation. This action should be called after the `CreateNew`, `SetMinimumConfidence`, and `SetSignatureReferenceFolderPath` actions.

Member of namespace

SignatureValidation

Syntax

```
bool ValidateSignature (bool createFields)
```

Parameter

`createFields`

True or False value indicating whether to add a new field to the current object for each signature that is detected in the document. Runs the signature detection and validation.

Details

Performs signature detection and validation. This action should be called after the `CreateNew`, `SetMinimumConfidence`, and `SetSignatureReferenceFolderPath` actions.

Note:

This action is supported at the page level only for images identified as US checks. If the page is not a US check, then the signature must be in a zoned field that contains only the signature. To determine if an image is a check, use the actions in the appropriate [check processing](#) library. It is possible that signature stamps might be validated by this action, however signature stamps are not supported.

After the action runs, variables are created at the calling DCO level when the following conditions are met:

1. A signature is detected and matched. When this condition occurs, the following variables are created:
 - *SignatureFound* is set to "1".
 - *SignatureConf* is set to an integer value representing the confidence of the match.
 - *SignatureReference* is set to a string value representing the name of the best matching signature reference.
 - *SignatureIndex* is set to an integer value representing the index of the best matching signature reference file.
 - *SignatureValidated* is set to "1".
2. A signature is detected, but not matched. When this condition occurs, the following variable is created:
 - *Signature_NotVerifiedX*, where X is the index of the signature that was detected but not validated.

Level

Page and field.

Returns

True, always.

```
SetSignatureReferenceFolderPath("@X.referenceFolderPath", "*" + @X.accountId + "*")
CreateNew("True", "@X.accountId")
SetMinimumConfidence("70")
ValidateSignature(False)
```

This example:

- sets the signature references directory folder path by using the `SetSignatureReferenceFolderPath` action, and compares only against the reference files with names that contain the value of a previous captured *accountId* variable.
- instructs the signature matching algorithm to create a new reference file if the signature is not matched, and to name it using the value of a previously captured *accountId* variable.
- sets the minimum confidence to 70 so that matching signatures with a confidence less than 70 are ignored.
- runs the signature detection/validation.

Parent topic: [SignatureValidation actions](#)

SPEXport actions

Use the SPEXport actions to upload documents to a Microsoft SharePoint library.

The SPEXport actions integrate Datacap applications with the SharePoint library. You run these actions to access the SharePoint server, set up document attributes and folders on the server, and upload documents to the server for storage.

- [SP_CreateFolder](#)
Creates the folder in the SharePoint where you import your documents.

- [SP_Login](#)
Creates the connection to SharePoint library by using the user ID, password, optional SharePoint domain.
- [SP_SetContentType](#)
Sets the type of content in the SharePoint library for the uploaded documents, such as an Invoice.
- [SP_SetFileType](#)
Sets the format in which to upload the document to the SharePoint library, for example TIF or PDF.
- [SP_SetProperty](#)
Sets the column property in SharePoint for the documents you want to upload.
- [SP_SetUploadMode](#)
Identifies the files to upload into the SharePoint library.
- [SP_SetUrl](#)
Sets the URL address of the SharePoint library.
- [SP_Upload](#)
Uploads the image file and any indexes that are specified for the batch, document, or page into SharePoint.
- [SP_UploadDir](#)
Uploads the file into the specified folder.

Parent topic: [Global actions](#)

SP_CreateFolder

Creates the folder in the SharePoint where you import your documents.

Syntax

```
bool SP_CreateFolder (StrParam)
```

Parameters

The URL that specifies the folder to create in SharePoint. Smart parameters are supported. Refer to the Smart Parameter documentation for more information.

Returns

True if the folder was created successfully or if it already exists. Otherwise, False.

Level

All.

Details

Creates the SharePoint folder specified in the parameter string.

Note: The SP_SetUrl action optionally can define directories and subdirectories to be created during the upload.

Example

```
SP_CreateFolder ("http://blue/Docs/Documents/Test")
```

Parent topic: [SPExport actions](#)

SP_Login

Creates the connection to SharePoint library by using the user ID, password, optional SharePoint domain.

Syntax

```
bool SP_Login(StrParam)
```

Parameters

A string containing 3 comma separated input parameters.

1. SharePoint userID.
2. Password.
3. An optional SharePoint domain. If not included, do not include the preceding comma.

Smart Parameters are supported. Use smart parameters to prevent clear text passwords in your application by obtaining the password from the application service.

Returns

True if the login succeeded. Otherwise, False.

Note: If the login parameters are invalid, a failure may not occur until you call SP_Upload.

Level

All.

Details

Login to SharePoint with credentials other than the logged-in Windows User.

Example

```
SP_SetUrl("http://blue/Docs/Documents/+#BatchID/+#ID")
SP_Login("userID,password,domain")
SP_SetContentType("Invoice")
SP_SetFileType("jpg")
SP_SetProperty("Date,@Value")
SP_Upload()
```

Alternatively, you can use smart parameters to obtain information from the application service to prevent clear text passwords. Here is an example where the password is stored in a custom value called SPPassword in the application service:

```
SP_Login("userID,@APPVAR(values/adv/SPPassword),domain")
```

Parent topic: [SPExport actions](#)

SP_SetContentType

Sets the type of content in the SharePoint library for the uploaded documents, such as an Invoice.

Syntax

```
bool SP_SetContentType (StrParam)
```

Parameters

A valid SharePoint content type in the selected Library. No error is raised if it is not a valid content type. Smart Parameters are supported.

Returns

True if the content type was successfully set or if the content type is not a valid content type. False if there is failure returned from SharePoint.

Level

All.

Details

This action sets the SharePoint Content Type for each document that is subsequently uploaded. Content Type is a SharePoint concept that defines a subset of columns (fields) within a library of documents, to be displayed and edited for a specific purpose.

Example

```
SP_SetContentType ("Invoice")
```

Parent topic: [SPExport actions](#)

SP_SetFileType

Sets the format in which to upload the document to the SharePoint library, for example TIF or PDF.

Syntax

```
bool SP_SetFileType (StrParam)
```

Parameters

A string indicating the type or filename extension of the images to be uploaded for each document or batch. When uploading the Batch or Document this extension is appended to the BatchID or DocumentID to select the image. The IMAGEFILE property takes precedence for Page uploads. See the description of SP_Upload for details. Valid parameters include: tif, tiff, jpg, jpeg, jpe, gif or pdf. The parameter may optionally include a period (for example .tif and .jpeg are also valid).

Smart parameters are supported.

Returns

False if the parameter is not a three-character extension, jpeg, or tiff, with or without a leading period. Otherwise True.

Note: If a three-character extension is supplied that is invalid for SharePoint images, the upload may fail.

Level:

All.

Details

Note: SP_SetUploadMode takes precedence over SP_SetFileType, if SP_SetUploadMode is called prior to SP_Upload this parameter has no effect. If neither SP_SetFileType nor SP_SetUploadMode are called, tif is used as the default file type.

Example

```
SP_SetFileType("jpg")
```

Parent topic: [SPExport actions](#)

Related reference:

[SP_Upload](#)

SP_SetProperty

Sets the column property in SharePoint for the documents you want to upload.

Syntax

```
bool SP_SetProperty(StrParam)
```

Parameters

Two comma separated values:

1. Column name is the name or ID of the target column in SharePoint.
2. Data value is the value to be uploaded to that column. Refer to the documentation for more information about the column types.

Smart Parameters are supported.

Returns

True if the parameters are not blank. The index information is uploaded to SharePoint when a document is subsequently uploaded. Otherwise, False.

Level

All.

Details

Sets an index value (column in SharePoint) for the documents to follow. Can be called multiple times to set multiple index values.

Notes®:

- Any spaces in column names must be replaced with “_0x02c_”.
- The real column name may be different from what is displayed in SharePoint. To determine the real Column name select the column settings and check the browser address.
- For example for the property called Description you may see "...3F2%7D Field=Comments" at the end. This means that the real name of the Column to be used in the SP_SetProperty action is "Comments".

Example

```
SP_SetProperty("Date,@Value")
SP_Upload()
```

Moves the value of the current field to the SharePoint column named Date.

Parent topic: [SPExport actions](#)

SP_SetUploadMode

Identifies the files to upload into the SharePoint library.

Syntax

```
bool SP_SetUploadMode (StrParam)
```

Parameters

A string or Smart Parameter identifying the page level variable where file name stored. If this action is not called the value defaults to blank and regular upload logic applied. For example `SP_SetUploadMode("ParentImage")` will cause uploading file with the name stored in *ParentImage* variable on the page level.

Returns

Always True.

Level

Batch, Document or Page level.

Details

Use this action to identify the files that will be uploaded to SharePoint.

Example

```
SP_SetUploadMode (ParentImage)
SP_Upload()
```

Parent topic: [SPExport actions](#)

SP_SetUrl

Sets the URL address of the SharePoint library.

Syntax

```
bool SP_SetUrl (StrParam)
```

Parameters

The full URL to the SharePoint repository. Smart parameters are supported. Refer to the Smart Parameter documentation for more information.

Returns

True if the action succeeded. Otherwise, False.

Level

All.

Details

Sets target URL of location to which image files are uploaded.

Note: /Docs/Documents/ is the default Document Library within SharePoint site.

Example:

```
SP_SetUrl("http://blue/Docs/Documents/>{@BatchID}/{@ID}")
```

With this example, directories with names defined by />{@BatchID}/{@ID} are created automatically during upload.

Parent topic: [SPExport actions](#)

SP_Upload

Uploads the image file and any indexes that are specified for the batch, document, or page into SharePoint.

Syntax

```
bool SP_Upload()
```

Parameters

None.

Returns

True if all documents and indexes were successfully uploaded. Otherwise, False.

Level

All.

Details

Uploads the image file and any indexes specified for the current page, document, or batch to SharePoint. Uses TiffMerge file naming scheme to find document level or batch level image file: DocID.TIF or BatchID.TIF by default. Pages associated with other image file types (e.g. TM000001.pdf, TM000001.jpg, etc) can be uploaded.

Note: After uploading, the variable *Upload_Folder* in the page/doc/batch will be set to the SharePoint URL where the document(s) were uploaded.

Note: If some documents in a batch are successfully uploaded and some fail, and the batch is rerun through the SharePoint Upload task, only documents that failed to upload will be re-uploaded.

Note: If any document is re-uploaded, SharePoint will replace the existing document with the newer one, or save the old version and replace it with the new version, depending on SharePoint Versioning settings.

Example

```
SP_Upload()
```

Parent topic: [SPExport actions](#)

SP_UploadDir

Uploads the file into the specified folder.

Syntax

```
bool SP_UploadDir(StrParam)
```

Parameters

Two comma separated parameters:

1. The Windows folder containing only document files to upload.
2. A Boolean. True means delete file after upload, false means move file to the Uploaded folder in current directory.

Smart parameters are supported.

Returns

True if the upload succeeds for all files in the directory. Otherwise, False.

Level

Batch or Document level.

Details

Uploads all files in specified folder.

Example

```
SP_UploadDir("C:\ParentDir\Invoice\Images\Input\", false)
```

Parent topic: [SPExport actions](#)

Split actions

Use the Split action to split a batch into smaller batches so each can be processed separately.

The Split action splits batches based on the value of the specified document-level variable.

- [SplitBatch](#)
Splits a batch into smaller batches that are based on the value of the specified document-level variable.

Parent topic: [Global actions](#)

SplitBatch

Splits a batch into smaller batches that are based on the value of the specified document-level variable.

Syntax

```
bool SplitBatch (StrParam)
```

Parameters

A smart parameter pointing to a Document or Page variable that determines if the Document or Page is to be split to a Child Batch.

Important: The action evaluates all documents and pages (including unbound* pages) in the batch.

The values of the smart parameter variable found during the document and page evaluation are grouped into buckets:

1. Pages/documents that contain the variable and the variable values are identical go into the same bucket.
2. If there are multiple buckets, all pages/documents that share the same value will split to the same child batch.
3. There can be only one child batch for each unique bucket value.

Child batches have the same name as the parent batch, but include an additional two character alpha-decimal suffix such as .01, .02, .0A, ..., up to .ZZ. This hexadecimal numbering for child batches is required by Datacap Server. Datacap Server creates the batch and queue entries for the child batches after the task is finished, when the split condition is processed. There is a maximum of 1295 child batches.

Example: @D.Inbox. If there is an Inbox variable in each document, this will split documents by the value of the Inbox variable.

Important: Any document or unbound* page that does not have this Inbox variable value, will remain in the parent batch.

*An unbound page is any page not inside a document.

Returns

False if an error occurs like a file could not be created, etc., and the batch will be set to abort. Otherwise True.

If the specified variable is not found in any documents or unbound pages, meaning there is nothing to split, the action is still considered to be successful and will return true.

Each child batch split off will generate a condition, which should be configured for Split in the workflow.

Any page or document with a blank value for the splitting value will remain in the original "parent" batch.

Level

Batch level only.

Details

This action will process all of the documents and unbound pages that are in the batch and attempt to split the identified documents and pages into child batches. The action will look in the documents or unbound pages for the variable specified in the `a` parameter, group the objects that have a matching value, and split each group into a unique child batch. Only documents and unbound pages will be processed. Pages that are already placed within in a document structure will not be processed individually, the pages will be as part of their document, not as a separate page.

Additional considerations:

There is only one job routing condition raised by this action: it is the first one in the task's list of conditions.

The task's Task Setup/Task Settings screen must be configured as Job Router, and a single condition defined (by convention, call it Split).

1. Any and all child batches will be routed via this single condition.
2. If the application wants to treat the individual buckets differently, then the first step in the workflow after splitting can check the same smart parameter value and branch or re-route the child batch using that value.
3. All the structure and variables, etc. that were in the parent batch docs/pages are retained in the child batches.
4. In addition, new variables *ParentBatch* and *ParentBatchDir* are added.
5. The action can only be used once per Parent Batch.
6. The maximum number of child batches is 1295.
7. The page count and document count in child batches is not accurate after splitting. It is updated and will be accurate once the next task completes.

Example

```
SplitBatch (@D.Inbox)
```

Parent topic: [Split actions](#)

Statistics actions

These statistics actions are used by the Profile Statistics and Export Statistics rulesets to save information about field recognition accuracy and page classification accuracy.

- [AddToDBTotals](#)
Updates the application summary statistics in the reportTotal table in the Engine database with the cumulative batch size total in KB (*rt_SizeKB*) and the cumulative total number of checks processed (*rt_Feature1Count*).
- [CompareFieldsText](#)
Calculates the page type classification and field recognition accuracy statistics and updates database report tables. Call this action after data verification to calculate accuracy based on changes to the data during verification.
- [IsBatchAborted](#)
Checks if the batch has already been aborted and stops running the ruleset if so.
- [SaveFieldsText](#)
Saves recognized page types and field values after recognition for accuracy calculations.

Parent topic: [Global actions](#)

AddToDBTotals

Updates the application summary statistics in the reportTotal table in the Engine database with the cumulative batch size total in KB (*rt_SizeKB*) and the cumulative total number of checks processed (*rt_Feature1Count*).

Syntax

```
bool AddToDBTotals (strParam)
```

Parameters

None

Returns

Always True

Level

Batch level only

Details

When this action runs, it also sets the *TotalStatsUpdated* batch variable to avoid double-counting.

Example

```
AddToDBTotals ()
```

Parent topic: [Statistics actions](#)

CompareFieldsText

Calculates the page type classification and field recognition accuracy statistics and updates database report tables. Call this action after data verification to calculate accuracy based on changes to the data during verification.

Syntax

```
bool CompareFieldsText (strParam)
```

Parameters

None

Returns

Always True

Level

Batch level only

Details

CompareFieldsText requires that you have previously run the SaveFieldsText action between the recognition (Profiler) and verification tasks. You can prevent inclusion of a page or field in accuracy statistics by setting the *ClassifiedPageType* or *RecognizedText* variables to an empty string before calling CompareFieldsText. The action will populate the following batch variables:

PageClassifiedErrors

The number of page classification errors (errors that result in a change to the page type after auto-classification).

PageClassifiedCount

The number of pages classified in the batch.

FieldRecognitionCount

The number of fields recognized.

FieldRecognitionErrors

The number of fields that were incorrectly recognized (fields that were changed after recognition).

Example

```
CompareFieldsText ()
```

Parent topic: [Statistics actions](#)

IsBatchAborted

Checks if the batch has already been aborted and stops running the ruleset if so.

Syntax

```
bool IsBatchAborted ()
```

Parameters

None

Returns

True if the batch is set to abort, otherwise False.

Level

Any

Example

```
IsBatchAborted ()
```

Parent topic: [Statistics actions](#)

SaveFieldsText

Saves recognized page types and field values after recognition for accuracy calculations.

Syntax

```
bool SaveFieldsText (strParam)
```

Parameters

None

Returns

Always True

Level

Batch level only

Details

Call this action after page classification and recognition and before verification to save the original recognized data for statistics calculations. The action will populate the following page and field variables:

ClassifiedPageType

Set for each page to the current page type

RecognizedText

Set for each non-empty field to the current value of the field.

Example

```
SaveFieldsText ()
```

Parent topic: [Statistics actions](#)

TifMerge actions

Use the TifMerge actions to combine individual TIFF images into a multi-page TIFF file. This action is typically run at the end of the workflow so that you can upload or release the batch images as a single file.

The TifMerge actions determine the path to the Batch directory, creates the multi-page file, and lets you specify the compression to use in the final image.

- [TifMerge_CheckStatus](#)
Filters merged pages and documents that are based on their DCO status.
- [TifMerge_ExportToBatchDir](#)
Indicates that the path for the multi-Image file is to the current Batch directory.
- [TifMerge_MergeImages](#)
Merges the images associated with the object of the Document Hierarchy to which the action's ruleset applies into a single, multi-Image file.
- [TifMerge_MyImage](#)
Adds each single image to the multi-Image file.
- [TifMerge_PreserveCompression](#)
Determines the output compression type for merged images.
- [TifMerge_SetFileName](#)
Sets the name of the multi-Image file (.tif) to be created by TifMerge.

- [TifMerge_SetFilePath](#)
Sets the path for the multi-Image file.

Parent topic: [Global actions](#)

TifMerge_CheckStatus

Filters merged pages and documents that are based on their DCO status.

Syntax

```
bool TifMerge_CheckStatus (string AcceptablePageStatuses, string  
DisregardPageStatuses, string AcceptableDocStatuses, string DisregardDocStatuses)
```

Parameters

AcceptablePageStatuses
Type: string

DisregardPageStatuses
Type: string

AcceptableDocStatuses
Type: string

DisregardDocStatuses
Type: string

Parameters

1. AcceptablePageStatuses: a comma-separated list of the Page status values that are merged.
2. DisregardPageStatuses: a comma-separated list of the Page status values that are not merged.
3. AcceptableDocStatuses: a comma-separated list of the Document status values that are merged.
4. DisregardDocStatuses: a comma-separated list of the Document status values that are not merged.

Returns

Always True

Level

Any

Details

This action configures the acceptable statuses for documents and pages when you call the `TifMerge_MergeImages` action. If this action is not called, the status values of documents and pages is not checked.

In the following example, only the pages with an acceptable status of '0' or '49' are merged. The disregard page status of '75' is redundant because of its exclusion from the acceptable status values. By contrast, the disregard document status of '128' prevents all of the child pages of the document from being merged.

Example

```
TifMerge_CheckStatus("0,49","75","","128")
TifMerge_MergeImages("All")
```

Parent topic: [TifMerge actions](#)

TifMerge_ExportToBatchDir

Indicates that the path for the multi-Image file is to the current Batch directory.

Syntax

```
()
```

Parameters

None.

Returns

False if the path does not exist or is not accessible. Otherwise, True.

Level

Batch or Document usually, but Page or Field is permissible.

Details

When saving a multi-image file, this action is used to configure the current batch directory as the destination for the output file. This action must be called before the action to merge the images.

Example

```
TifMerge_ExportToBatchDir()
TifMerge_MergeImages("All")
```

Parent topic: [TifMerge actions](#)

TifMerge_MergeImages

Merges the images associated with the object of the Document Hierarchy to which the action's ruleset applies into a single, multi-Image file.

Syntax

```
bool TifMerge_MergeImages (sPageType)
```

Parameters

String value indicating either:

1. All if the multi-Image file is to contain images of all pages without regard to Page Type.
2. The Page Type(s) of the images to be included (comma-separated list, if the parameter includes more than one Page Type.)

Smart parameters are supported.

Returns

False if the action cannot create the multi-Image file. Otherwise, True.

Level

Batch or Document.

Details

This action merges the images associated with the object to which the action's rule applies into a single, multi-Image file.

At the Batch level, the action merges all Image files in the batch into one multi-Image file – or those Image files representing pages of the Page Type you specify as a parameter. At the Document level, the action assembles a new multi-Image file for each document in the batch. If you specify a Page Type, the multi-Image file for each document will include only images of pages of that type.

Actions `TifMerge_SetFileName` and `TifMerge_SetFilePath` must be called before `TifMerge_MergeImages`.

Example

```
TifMerge_SetFileName("@BATCHID+@DATE(dd.mm.yyyy)")
TifMerge_SetFilePath("C:\ParentDir\Invoice\batches\MultiImage")
TifMerge_MergeImages("All")
```

```
TifMerge_SetFileName("Doc_+@ID+@DATE(dd.mm.yyyy)")
TifMerge_SetFilePath("C:\ParentDir\Invoice\MultiImage")
TifMerge_MergeImages("Invoice,Attachment")
```

The first example merges all images into a multi-Image file that uses the Batch ID and the processing Date for its name.

The second example applies to a Document object of the Document Hierarchy. It assembles a multi-Image file for each document in the batch; the images in a file are limited to Invoice and Attachment pages.

Parent topic: [TifMerge actions](#)

TifMerge_MyImage

Adds each single image to the multi-Image file.

Syntax

```
bool TifMerge_MyImage ()
```

Parameters

None.

Returns

False if the action's ruleset is not applied to a Page object or if the corresponding image file for the current page cannot be found. Otherwise, True.

Level

Page level.

Details

Adds the current page image to the multi-Image TIF file. Actions that proceed the TifMerge_MyImage action allow you to specify under which conditions an image is to be added to the multi-Image file.

A rule with this action can only be applied to a Page object. The output image file name and destination path must have been previously set using TifMerge_SetFileName and TifMerge_SetFilePath.

Example

```
ChkDDCOStatus ("0")  
TifMerge_MyImage ()
```

In this example, the ChkDDCOStatus action checks that the status of the current page is "0". If so, the image for the current page will be added to the multi-Image file.

If the current status of this page is not "0", the rule will fail and the TifMerge_MyImage action will not be run; therefore, the current image will not be added to the multi-Image file.

This ChkDDCOStatus action is used as an example. You can use many other actions to be sure the current image meets your merging criteria.

Parent topic: [TifMerge actions](#)

Related reference:

[TifMerge_SetFileName](#)

[TifMerge_SetFilePath](#)

TifMerge_PreserveCompression

Determines the output compression type for merged images.

Syntax

```
bool TifMerge_PreserveCompression (string PreserveCompression)
```

Parameters

string PreserveCompression

Parameters

True: Preserves the original compression type of the source image.

False: Uses G4 compression for black and white images. JPEG is used for color images.

Returns

Always True.

Level

Any.

Details

This action will configure the output format for TifMerge_MergeImages. If this action is not called, then the default value of False is used, so the original image compression is not preserved.

Example

```
TifMerge_PreserveCompression("TRUE")
TifMerge_MergeImages("All")
```

Parent topic: [TifMerge actions](#)

TifMerge_SetFileName

Sets the name of the multi-Image file (.tif) to be created by TifMerge.

Syntax

```
bool TifMerge_SetFileName (StrParam)
```

Parameters

String value of the file name to be assigned to the multi-page file. Smart parameters are supported.

Returns

Always True.

Level

All.

Details

This action sets the name of the multi-Image file (.tif) to be created by the TifMerge actions. The file name can be text, or a combination of text and the value of a variable you enter as a parameter. The action automatically adds the “.tif” extension to the file.

Example

The following example assumes that the rules with these actions are applied to a Document object. The names combine text values such as “Doc_” with values assigned to variables by using smart parameters.

```
TifMerge_SetFileName("Doc_+@ID+@DATE(dd.mm.yyyy)")
TifMerge_SetFilePath("c:\ParentDir\Invoice\MultiImage")
TifMerge_MergeImages("All")
```

This example combines "MultiTif_" with the ID of the Document Hierarchy object to which the ruleset is applied. Usually, a rule that contains this action applies to a Batch object or Document object, but can apply to a Page or Field object.

```
TifMerge_SetFileName ("MultiTif_+@ID")
```

Parent topic: [TifMerge actions](#)

Related reference:

[TifMerge_SetFilePath](#)

TifMerge_SetFilePath

Sets the path for the multi-Image file.

Syntax

```
bool TifMerge_SetFilePath (strParam)
```

Parameters

String value for the output path of the multi-Image TIF file. Smart parameters are supported.

Returns

False if the specified drive does not exist or the path cannot be created. Otherwise, True.

Level

All.

Details

Sets the path for where the multi-Image file will be created. If the folder designated in the parameter does not exist, the action will create the folder in which the TIF file will be placed.

Usually, a rule containing this action applies to a Batch object or Document object of the Document Hierarchy, but can apply to a Page or Field object.

Example

```
TifMerge_SetFileName ("Doc_+@ID+@DATE (dd.mm.yyyy) ")  
TifMerge_SetFilePath ("c:\ParentDir\Invoice\MultiImage")  
TifMerge_MergeImages ("All")
```

Parent topic: [TifMerge actions](#)

Related reference:

[TifMerge_SetFileName](#)

TM524 actions

The TM524 actions are for compatibility with older versions of Datacap and are no longer used

Parent topic: [Global actions](#)

Validations actions

Use the Validations actions to check and modify the content and format of the current field value.

Other actions in the Validations library do arithmetic calculations, assign values, copy values, and check variables.

The Validations actions are described in the following table.

- [AddLeadingZeros](#)
Inserts zeros at the beginning of a value so the character count equals the number that is specified.
- [AddPaddingToEnd](#)
Pads the captured value of the current Field object with spaces from after the last character in the string out to the number of specified characters.
- [AddPaddingToStart](#)
Pads the captured value of the current Field object with spaces from the start of the string up to the first character until the specified length is reached.
- [AddTrailingZeros](#)
Adds zeros to the end of captured value of the current Field until the length of the value reaches the maximum *n* you enter as the parameter.
- [AllowOnlyChars](#)
Removes all of the characters that are not specified as supported.
- [AppendFromField](#)
Appends the captured value of the specified Field object to the captured value of the current Field object.
- [AppendToField](#)
Appends the captured value of the current Field object to the captured value of the Field object that is specified by the parameter.
- [AssignFieldDefault](#)
Assigns a default value to the current field.
- [CalculateDateDifference](#)
Calculate the differences between two dates and stores the calculation in a user defined variable.
- [CalculateFields](#)
Evaluates a specified expression as either True or False.
- [CheckSubFields](#)
Determines whether the values of the specified child fields meet the specified criteria and deletes the parent field if they do not.
- [CompareFields](#)
Compares the values of two fields by using the specified matching criteria that supports fuzzy matching.
- [ConvertFieldToCurrency](#)
Converts the value of the current field to a currency value.
- [ConvertToLowerCase](#)
Converts any upper case characters in the captured value of a Field object to lower case characters.
- [ConvertToUpperCase](#)
Converts the lower case characters in the captured value of a Field object to upper case characters.
- [CopyField](#)
Copies the value of the current field to a specified field.
- [CopyFieldToField](#)
Copies the value of the current field to a specified field.
- [DateStampField](#)
Updates the current Field object with today's date.
- [DeleteAllAlpha](#)
Deletes all of the alphabetic characters from the captured value of the current Field object.
- [DeleteAllMiscChars](#)
Deletes all of the UNICODE Symbol Category characters from the captured value of the current Field object.
- [DeleteAllNumeric](#)
Deletes all of the numeric characters from the captured value of the current Field object.

- [DeleteAllPunct](#)
Deletes all of the punctuation from the value of the current field.
- [DeleteAllSysChars](#)
Deletes all of the characters with ASCII values 0 through 31 from the captured value of the current Field object.
- [DeleteChildType](#)
Deletes all of the child objects of the type that you designate as a parameter from the Document Hierarchy.
- [DeleteLCSpaces](#)
Deletes all of the low confidence spaces from the value of the current field.
- [DeleteParentObj](#)
Deletes the parent of the Document Hierarchy object to which a rule that contains this action is bound.
- [DeleteSelectedChars](#)
Deletes a specified character sequence from the value of the current field. This action is a more flexible version of the FilterFieldSelectedChars action.
- [EmptyFieldValue](#)
Clears the text value in the field represented by the Field object of the Document Hierarchy that is specified by the parameter.
- [FieldContainsValue](#)
Determines whether the current field value contains some or all of the specified text but no additional text.
- [FilterFieldSelectedChars](#)
Deletes the specified characters from the value of the current field.
- [FormatNumberToLocale](#)
Evaluates the current field value for known number patterns and if a known pattern is detected, updates the decimal and digit separators characters to match that of the current locale.
- [GetJobID](#)
Assigns the current job ID to the Text property of the current object.
- [HasChildOfType](#)
Determines whether the current object has a child of the specified type.
- [InsertChars](#)
Inserts one or more characters into the value of the current field.
- [InsertDecimalPoint](#)
Inserts a decimal point into the value of the current field at the specified position.
- [IsFieldCurrency](#)
Determines if the captured value of the Field meets the currency format of the current locale.
- [IsFieldDate](#)
Checks that the value of the field has an acceptable Date format. This action uses the current locale setting to determine valid patterns.
- [IsFieldDateEqualOrAfter](#)
Checks that the Date value in the current field represented by the bound Field object of the Document Hierarchy is greater than or equal to the Date value in the field that is specified as the parameter.
- [IsFieldDateEqualOrBefore](#)
Checks that the date in the current field represented by the bound Field object of the Document Hierarchy is less than or equal to the Date value in the field that is specified as the parameter.
- [IsFieldDateUpToToday](#)
Checks that the Date value of the current Field object is today's date or earlier.
- [IsFieldDateWithinRange](#)
Checks that the value assigned to the Text property of the bound object is a valid Date. If yes, this action then confirms that the Date is within the range specified by the parameters.
- [IsFieldDateWithinXDays](#)
Checks that the captured Date value of the current Field object is within *n* days of the number entered as a parameter.

- [IsFieldDateWithReformat](#)
Confirms that the data of a field is a valid date and then formats the Date value according to the format entered as the parameter.
- [IsFieldEmpty](#)
Checks that the Field object designated as a parameter does not have a captured value.
- [IsFieldFilled](#)
Determines whether the Field object designated as a parameter contains a captured value or is empty.
- [IsFieldGreaterOrEqual](#)
Determines if the captured value of the current Field object is greater than or equal to the value entered as a parameter.
- [IsFieldHidden](#)
Checks that the calling field is Hidden, which means the field has a variable *STATUS* that is equal to -1.
- [IsFieldLengthMax](#)
Checks that the character length of the current Field object's captured value is equal to or less than the value set as a parameter.
- [IsFieldLengthMin](#)
Checks the character length of the current Field object's captured value to see if its length is equal to or longer than a number *n*.
- [IsFieldLessOrEqual](#)
Determines if the captured value of the current Field object is less than or equal to the value entered as a parameter.
- [IsFieldMatching](#)
Determines if the value entered as the parameter is identical to the captured value of the current Field object.
- [IsFieldPercentAlpha](#)
Determines if the characters in the captured value of current Field object are *n%* alphabetic.
- [IsFieldPercentNonNumeric](#)
Determines if any of the characters in the captured value of the current Field object are *n%* not numeric characters.
- [IsFieldPercentNumeric](#)
Determines if the characters in the captured value of the current Field object are *n%* numeric characters.
- [IsMatchingJobID](#)
Checks that the Job ID of the current User Application job matches the Job ID value of the parameter.
- [IsMaxOMRChecked](#)
Indicates the maximum number of check boxes that can contain a value, such as a check.
- [IsMinOMRChecked](#)
Indicates the minimum number of check boxes that can contain a value, such as a check.
- [IsPatternInField](#)
Checks that the value of the current field contains the specified VBScript regular expression.
- [IsSupportedImageFile](#)
Checks that the image file attached to the current page is in a supported image format.
- [IsThisFieldEmpty](#)
Checks that the value of the current field is empty.
- [IsThisFieldFilled](#)
Checks that the current field has a captured value.
- [IsVariableEmpty](#)
Checks that the variable specified by the parameter does not contain a value.
- [IsVariableFilled](#)
Checks that the variable specified by the parameter contains a value.
- [ParseMultilineAddress](#)
Splits the value of the current field at each comma and saves the substrings to the specified fields. Typically used for address fields.

- [ParseName](#)
Splits the three word value of the current field and saves the substrings to the specified fields. Typically used for name fields, such as first name, middle name/initial, last name.
- [ReplaceChars](#)
Replaces a character or string of characters in the captured value of the current Field object with a String that you enter as one of the parameters.
- [ReplaceValueAtPosition](#)
Replaces the value at the specified position within the current field with a replacement string, or deletes the value.
- [ResetField](#)
Deletes the value of the current field and sets the Position attribute of the field to 0,0,0,0.
- [SetIsOverrideable](#)
Specifies if the user can or cannot override a validation that fails for the current object.
- [SplitFieldValuePreserveEnd](#)
Splits the captured value of a Field object at the first instance of the character specified as a parameter.
- [SplitFieldValuePreserveStart](#)
Splits the captured value of a Field object at the first instance of the character specified as a parameter.
- [SumFields](#)
Adds the values of all child fields of the specified type and assigns the result to the current field. You can also use this action to sum the values of the specified variable for all child objects.
- [TimeStampField](#)
Updates the current Field object with the current time.
- [TrimSpaces](#)
Deletes extra spaces at the beginning and end of the captured value of the current Field object.
- [TruncateFromEnd](#)
Deletes characters from the end of the captured value of the current Field object until the length of the value equals the length indicated by the parameter.
- [TruncateFromStart](#)
Deletes characters from the start of the captured value of the current Field object until the length of the value equals the length specified by the parameter.

Parent topic: [Global actions](#)

AddLeadingZeros

Inserts zeros at the beginning of a value so the character count equals the number that is specified.

Syntax

```
bool AddLeadingZeros (strParam)
```

Parameters

A number n which is the maximum length of the value. Smart parameters are supported.

Returns

False if the parameter you enter is not numeric. Otherwise, True.

Level

Field level.

Details

Adds zeros ("0") to the beginning of the captured value of the current Field object until the total length of the value reaches the maximum *n* you specify as the parameter.

Example

```
AddLeadingZeros("10")
2240.00 becomes 0002240.00
```

Parent topic: [Validations actions](#)

Related reference:

[AddTrailingZeros](#)

[IsFieldLengthMax](#)

[AddPaddingToLeft \(deprecated\)](#)

AddPaddingToEnd

Pads the captured value of the current Field object with spaces from after the last character in the string out to the number of specified characters.

Syntax

```
bool AddPaddingToEnd (strParam)
```

Parameters

A number *n* indicating the maximum permissible length of the value. If the action finds that a value's length is less than this number, it will insert spaces until the maximum length is reached. Smart parameters are supported.

Returns

Always True.

Level

Field level.

Details

Example

```
AddPaddingToEnd("10") uses spaces to expand a value with less than 10 characters. For example:
456.11 becomes 456.11_ _ _ _
```

Parent topic: [Validations actions](#)

Related reference:

[AddPaddingToStart](#)

AddPaddingToStart

Pads the captured value of the current Field object with spaces from the start of the string up to the first character until the specified length is reached.

Syntax

```
bool AddPaddingToStart (strParam)
```

Parameters

A number n indicating the maximum permissible length of the value. If the action finds that a value's length is less than this number, it inserts spaces until the maximum length is reached. Smart parameters are supported.

Returns

Always True.

Level

Field level.

Details

Example

```
AddPaddingToStart("12") uses spaces to expand a value with less than 12 characters.
```

```
For example: RSJ-112 becomes _ _ _ _ _RSJ-112
```

Parent topic: [Validations actions](#)

Related reference:

[AddPaddingToEnd](#)

AddTrailingZeros

Adds zeros to the end of captured value of the current Field until the length of the value reaches the maximum n you enter as the parameter.

Syntax

```
bool AddTrailingZeros (strParam)
```

Parameters

A number n which is the maximum length of the value. Smart parameters are supported.

Returns

False if the parameter you enter is not numeric; otherwise, True.

Level

Field level.

Details

Example

```
AddTrailingZeros("10")  
2240.00 becomes 2240.00000
```

Parent topic: [Validations actions](#)

Related reference:

[AddLeadingZeros](#)

AllowOnlyChars

Removes all of the characters that are not specified as supported.

Syntax

```
bool AllowOnlyChars (StrParam)
```

Parameters

A Regular Expression that specifies permitted characters in the current word.

Returns

Always True.

Level

Field level.

Details

This action employs a Regular Expression as its parameter to identify and remove all of the characters that are not in the parameter from the value of the Field.

An empty argument removes all characters.

Example

```
AllowOnlyChars("ABCDEFG.")  
HELLO DOLLY. becomes ED.
```

Parent topic: [Validations actions](#)

AppendFromField

Appends the captured value of the specified Field object to the captured value of the current Field object.

Syntax

```
bool AppendFromField (strParam)
```

Parameters

The name of the Field object whose text value is to be appended to the value of the current field.

Returns

False if the parameter is not the name of a Field object. Otherwise, True.

Level

Page or Field Level.

Details

You can also apply the action at the Page level. A Text page-level variable with the appended value is added to the Data file of the page.

Example

```
AppendFromField("Number")
```

Parent topic: [Validations actions](#)

AppendToField

Appends the captured value of the current Field object to the captured value of the Field object that is specified by the parameter.

Syntax

```
bool AppendToField (strParam)
```

Parameters

The name of the Field object to which the value is to be appended.

Returns

False if the parameter is not the name of a Field object. Otherwise, True.

Level

Field level.

Details

Example

```
AppendToField("FirstName")
```

If the current Field object is MiddleInitial, a rule with this action appends the value of the Middle Initial field to the FirstName Field object.

Parent topic: [Validations actions](#)

AssignFieldDefault

Assigns a default value to the current field.

Syntax

```
bool AssignFieldDefault (StrParam)
```

Parameters

The String value you're assigning to the field.

Returns

False if not called on the correct level. Otherwise, True.

Level

Field level.

Details

Example

```
AssignFieldDefault("Bill Paid")  
or  
AssignFieldDefault("PastDue!")
```

Parent topic: [Validations actions](#)

CalculateDateDifference

Calculate the differences between two dates and stores the calculation in a user defined variable.

Syntax

```
bool CalculateDateDifference (string startDate, string endDate, string  
targetVariable, string dateProperty)
```

Parameters

- startDate : The starting date
- endDate : The ending date
- targetVariable : The variable to hold the calculated result value
- dateProperty : The value to calculate, 0 = days, 1 = months, 2 = quarters, 3 = years.

Smart parameters are supported.

Returns

False if the format of either date is invalid. Otherwise, True.

Level

Any level.

Details

Calculates the number of days, months, quarters or years between two dates. Only whole numbers are returned. Any fractional parts of the value are dropped. Quarters are calculated simply by dividing the number of months by 3. The order of the dates does not matter.

This action only supports Gregorian short dates as input and the date format must match the default format of the current locale.

Example

```
CalculateDateDifference("4/20/2012", "5/19/2012", "@P.Months", 1)
```

This example creates the page variable *Months* with a value of 0.

```
CalculateDateDifference("4/20/2012", "5/20/2012", "@P.Months", 1)
```

This example creates the page variable *Months* with a value of 1.

```
CalculateDateDifference("4/20/2012", "4/19/2013", "@P.Years", 3)
```

This example creates the page variable *Years* with a value of 0.

```
CalculateDateDifference("4/20/2012", "4/20/2013", "@P.Years", 3)
```

This example creates the page variable *Years* with a value of 1.

```
CalculateDateDifference(@P\MyDate1, @P\MyDate2, "@P.Days", 0)
```

This example creates the page variable *Days* with the number of days between the dates specified by the values in field MyDate1 and field MyDate2.

```
CalculateDateDifference(@P.MyDate1, @P.MyDate2, "@P.Days", 0)
```

This example creates the page variable *Days* with the number of days between the dates specified by page variables MyDate1 and MyDate2.

Parent topic: [Validations actions](#)

CalculateFields

Evaluates a specified expression as either True or False.

Syntax

```
bool CalculateFields (string Equation, string DecimalDigits, string PreserveStatus)
```

Parameters

Equation

The expression to evaluate. An expression consists of two or more subexpressions as defined by comparison operators (such as “=” or “>”). For example, if an expression has one comparison operator, it has two subexpressions: one subexpression to the left of the comparison operator and one subexpression to the right of the comparison operator. If an expression has two comparison operators, it has three subexpressions. And so on.

Each subexpression consists of one or more terms that are separated by arithmetic operators (+, -, *, /, ^). A term can be a numeric constant or the name of a field object. The fields that a subexpression references are called *dependent fields*. Surround any dependent field names with single quotation marks (').

Any dependent field with a value of null is treated as if the field had a value of zero.

Restriction: A locale mismatch can prevent this action from properly evaluating the expression. A mismatch occurs if the value of a dependent field is not formatted for that field's locale. For information about setting the locale, see [Setting locale values](#).

DecimalDigits

Optional. The number of decimal places to use for comparing the numerical results of subexpressions.

PreserveStatus

Optional. There are two possible values that determine whether the validation status of dependent fields is updated:

"True"	The validation status of dependent fields is not updated.
"False"	<p>The validation status of dependent fields is updated in the following manner:</p> <ul style="list-style-type: none">• If the expression evaluates to False, the validation status of dependent fields is updated to a failed status value. For information about validation failures, see Show validation failures to an operator.• If the expression evaluates to True, the validation status of dependent fields is updated to a done status value. <p>Important: The validation status of dependent fields could be updated by other rules and actions. When you configure your application, sequence your validation logic to work in tandem with this action.</p>

The default value is "False".

Returns

True if the expression can be evaluated and the expression evaluates to True. Otherwise, the action returns False.

For example, the expression cannot be evaluated if a field's value is not numeric.

Level

Field level.

Details

Example

Code	Comment
<code>CalculateFields("'SubTotal' + 'Shipping' + 'Tax' = 'Total'")</code>	
<code>CalculateFields("('SubTotal' + 'Shipping' + 'Tax') - '0.05' = 'Total'")</code>	

Code	Comment
<pre>CalculateFields("('Wages' + 'Interest' + 'Unemployment')>=('Gross'-.05')") CalculateFields("('Wages' + 'Interest' + 'Unemployment')<=('Gross'+.05')")</pre>	Call this action twice to validate that the result is within a range.

Parent topic: [Validations actions](#)

CheckSubFields

Determines whether the values of the specified child fields meet the specified criteria and deletes the parent field if they do not.

Syntax

```
bool CheckSubFields (StrParam)
```

Parameters

An expression that specifies which LINEITEM child fields are to be checked for the presence or absence of captured values. Within the expression, the name of each child Field object must be surrounded with single quotation marks ('). You can also use parentheses () in your expression. Action is placed on the parent of the LINEITEM field.

Returns

Always True.

Level

Field level.

Details

Validates an instance of a parent Field object by confirming the presence (or absence) of captured values for one or more of its child fields. (In the Invoices application, as an example, child fields of the LINEITEM parent include: ItemID, ItemDesc, Quantity, Price and LineTotal.)

Invalid parent Field objects are deleted if they do not meet this criteria.

This action usually runs in its own RuleSet (for example, in a Filter RuleSet) and would be applied to the DETAILS Field object (parent object to the LINEITEM field) in the Invoices example.

Example 1

```
Example 1: The captured values for a LINEITEM field are:
ItemID = 12345
ItemDesc =
Price = 12.00
LineTotal =
```

```
The action's parameter contains this expression:
CheckSubFields("('ItemID' OR 'ItemDesc') AND ('Price' OR 'LineTotal')")
```

In this example the action returns True and the current LINEITEM object is Valid.

Example 2

The captured values for the LINEITEM field are:
ItemID = 12345
ItemDesc = Thank you for your order
Price =
LineTotal =

The action's parameter contains this expression:
CheckSubFields("('ItemID' OR 'ItemDesc') AND ('Price' OR 'LineTotal')")

In this example the action's equation returns is False and the current LINEITEM object is Invalid. The field and the values of its child fields are deleted from the Data file.

Example 3

The captured values for the LINEITEM field are:
ItemID = 12345
LineTotal = gonetolunch

The action's parameter contains this expression:
CheckSubFields("('ItemID') AND ('LineTotal')")

In this example, the action returns True and validates the LINEITEM field, despite the presence of a nonsense entry in the LineTotal child field.

Parent topic: [Validations actions](#)

CompareFields

Compares the values of two fields by using the specified matching criteria that supports fuzzy matching.

Syntax

```
bool CompareFields (StrParam)
```

Parameters

Five comma-separated values:

1. String value of the source field's name. This is the field with a value to be compared.
2. String value of the target field's name. This is the field with a value to be compared to.
3. String value: Y or Yes; N or No. Alternatively, a Numeric value: 0 = No or 1=Yes. Yes (or Y or 1) allows the action to carry out a fuzzy rather precise comparison of the fields' values.
4. Numeric value of the percentage of required precision. Numbers less than 100 permit increasing fuzziness.
5. String value: Y or Yes; N or No. Alternatively, a Numeric value: 0 = No or 1=Yes. Yes (or Y or 1) directs the action to compare values in the fields word-by-word.

Returns

False if the designations of Field objects in the first two parameters are not valid, or if the Data Types of the values in the first two fields do not match. Otherwise, True.

Level

All.

Details

Locates values in two fields specified by the first two parameters. If values are present in both fields, the action compares the values according to the matching criteria of the last three parameters.

Example

```
CompareFields("Invoice_Date, Due_Date, Yes, 100, Yes")
```

Parent topic: [Validations actions](#)

ConvertFieldToCurrency

Converts the value of the current field to a currency value.

Syntax

```
bool ConvertFieldToCurrency ()
```

Parameters

None.

Returns

True if the text value is numeric and greater than one character. Otherwise, False.

Level

All levels, but generally applied at the Field level.

Details

Formats the text value of a field as a currency value. The following steps are performed on the field:

1. Removes existing currency symbols.
2. Replaces negative value characters such as parenthesis, 'NEG', 'CR' and trailing hyphen with a leading hyphen.
3. If a decimal exists, its position is not changed.
4. If a decimal does not exist and the field contains 2 or more characters, a decimal is inserted before the last two characters in the field.
5. If a decimal does not exist and the field contains less than 2 characters, no decimal is inserted.

Example

```
ConvertFieldToCurrency()  
A value of 1 remains 1  
A value of 12 becomes .12  
A value of 105 becomes 1.05  
A value of 104.009 remains 104.009
```

Parent topic: [Validations actions](#)

ConvertToLowerCase

Converts any upper case characters in the captured value of a Field object to lower case characters.

Syntax

```
bool ConvertToLowerCase ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Example

```
ConvertToLowerCase ()
```

To ensure that the characters in all Product ID's are lower case, a Validate rule that applies to a Document Hierarchy's Item Field object would include this action.

Parent topic: [Validations actions](#)

Related reference:
[ConvertToUpperCase](#)

ConvertToUpperCase

Converts the lower case characters in the captured value of a Field object to upper case characters.

Syntax

```
bool ConvertToUpperCase ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Example

```
ConvertToUpperCase ()
```

A Validate rule with this action, if applied to a State Field object which accepts only abbreviations, would be sure the captured values contain Upper Case letters (AZ, AL, etc.)

Parent topic: [Validations actions](#)

Related reference:

[ConvertToLowerCase](#)

CopyField

Copies the value of the current field to a specified field.

Syntax

```
bool CopyField (strParam)
```

Parameters

The name of the target Field object.

Returns

False if the parameter does not match the name of a Field object. Otherwise, True.

Level

Field level.

Details

Assigns the captured value of the current Field object to a sibling Field object you specify as the parameter.

Example

```
CopyField("Date")
```

This action places the captured value of the current field into the Date field.

Parent topic: [Validations actions](#)

CopyFieldToField

Copies the value of the current field to a specified field.

Syntax

```
bool CopyFieldToField (strParam)
```

Parameters

The name of the target Field object of the Document Hierarchy.

Returns

False, if the parameter does not match the name of a Field object. Otherwise, True.

Level

Field level.

Details

Copies the captured value of the current Field object to the Field object designated as the parameter of the action.

Example

```
If the current field's value is "1/1/05"
```

```
CopyFieldToField("Date")
```

```
assigns "1/1/05" (without the quotation marks) to the Document Hierarchy's Date field.
```

Parent topic: [Validations actions](#)

Related reference:

[CopyField](#)

DateStampField

Updates the current Field object with today's date.

Syntax

```
bool DateStampField (StrParam)
```

Parameters

A Date format such as mm/dd/yy or dd/mm/yy. (* defaults to mm/dd/yyyy)

Smart parameters are supported.

Returns

Always True.

Level

Field level.

Details

Example

```
DateStampField("*") produces  
01/20/2005
```

```
DateStampField("dd/mm/yy") produces  
20/01/05
```

Parent topic: [Validations actions](#)

DeleteAllAlpha

Deletes all of the alphabetic characters from the captured value of the current Field object.

Syntax

```
bool DeleteAllAlpha ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Example

```
DeleteAllAlpha ()  
JAN2003 becomes 2003
```

Parent topic: [Validations actions](#)

Related reference:

[DeleteSelectedChars](#)
[DeleteAllMiscChars](#)
[DeleteAllNumeric](#)
[DeleteAllPunct](#)

DeleteAllMiscChars

Deletes all of the UNICODE Symbol Category characters from the captured value of the current Field object.

Syntax

```
bool DeleteAllMiscChars ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Example

```
DeleteAllMiscChars ()  
  
'100c = $1' becomes '100 1'
```

Parent topic: [Validations actions](#)

Related reference:

[DeleteSelectedChars](#)

[DeleteAllAlpha](#)

[DeleteAllNumeric](#)

DeleteAllNumeric

Deletes all of the numeric characters from the captured value of the current Field object.

Syntax

```
bool DeleteAllNumeric ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Example

```
DeleteAllNumeric ()  
  
JAN2003 becomes JAN
```

Parent topic: [Validations actions](#)

Related reference:

[DeleteSelectedChars](#)
[DeleteAllAlpha](#)
[DeleteAllMiscChars](#)
[DeleteAllPunct](#)

DeleteAllPunct

Deletes all of the punctuation from the value of the current field.

Syntax

```
bool DeleteAllPunct ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Removes all characters with ASCII values 33-47,58-64,91-96, and 123-191 from the captured value of the current Field object.

Example

```
DatePUNCT ()
```

Parent topic: [Validations actions](#)

Related reference:

[DeleteSelectedChars](#)

[DeleteAllAlpha](#)

[DeleteAllMiscChars](#)

[DeleteAllNumeric](#)

DeleteAllSysChars

Deletes all of the characters with ASCII values 0 through 31 from the captured value of the current Field object.

Syntax

```
bool DeleteAllSysChars ()
```

Parameters

None.

Returns

Always True.

Level

Field only.

Details

Example

```
DeleteAllSysChars()  
A field containing Hello Dolly becomesHelloDolly.
```

Parent topic: [Validations actions](#)

DeleteChildType

Deletes all of the child objects of the type that you designate as a parameter from the Document Hierarchy.

Syntax

```
bool DeleteChildType (StrParam)
```

Parameters

String value of the child object Type.

Field, for example, eliminates all Field objects. In the Invoices application, LineItem removes the child fields of the Details parent Field object.

Returns

False if the child objects do not exist. Otherwise, True.

Level

All except Batch.

Details

Example

```
DeleteChildType ("Field")
```

Parent topic: [Validations actions](#)

DeleteLCSpaces

Deletes all of the low confidence spaces from the value of the current field.

Syntax

```
bool DeleteICSpaces ()
```

Parameters

None.

Returns

False if the action is unable to isolate the spaces in a field's value. Otherwise, True.

Level

All levels, but generally at the Field level.

Details

Uses the *ReqConf Setup dco* variable as threshold for low confidence values; defaults to 9.

Example

```
DeleteICSpaces ()
```

Parent topic: [Validations actions](#)

DeleteParentObj

Deletes the parent of the Document Hierarchy object to which a rule that contains this action is bound.

Syntax

```
bool DeleteParentObj ()
```

Parameters

None.

Returns

False if a parent or grandparent object cannot be found, or if the deletion of the parent object fails. Otherwise, True.

Level

Document, Page, and Field.

Details

Example

```
DeleteParentObj ()
```

DeleteSelectedChars

Deletes a specified character sequence from the value of the current field. This action is a more flexible version of the FilterFieldSelectedChars action.

Syntax

```
bool DeleteSelectedChars (strParam)
```

Parameters

strParam

A comma-delimited string that consists of the following parts:

Deletion string	The string that is to be deleted from the current field value. This string can consist of one character. Smart parameters are supported. To specify a comma as part of the deletion string, use the smart parameter @CHR as shown in the example.
Segment position	The position within the current field value that is the beginning of the string segment in which deletion is to occur. The default segment position is 1.
Instance maximum	The maximum number of deletion string instances that are to be deleted. To delete all instances, specify an asterisk ("*"). The default instance maximum is 1.

Returns

Always True.

Level

Field level.

Details

Example

```
DeleteSelectedChars("-", , "*")  
"223-56-7669" becomes "223567669"
```

```
DeleteSelectedChars("-", 5, "*")  
"223-86-7669" becomes "223-867669"
```

```
DeleteSelectedChars("@CHR(44), , 2")  
"Hello, Roger, Tom, and Jan" becomes "Hello Roger Tom, and Jan"
```

```
DeleteSelectedChars("er+@CHR(44), , *")  
"Hello, Roger, Tom, and Jan" becomes "Hello, Rog Tom, and Jan"
```

Related reference:[DeleteAllAlpha](#)[DeleteAllMiscChars](#)[DeleteAllNumeric](#)[DeleteAllPunct](#)

EmptyFieldValue

Clears the text value in the field represented by the Field object of the Document Hierarchy that is specified by the parameter.

Syntax

```
bool EmptyFieldValue (strParam)
```

Parameters

The name of the Field object that is to be emptied.

Returns

False if the field specified by the parameter does not exist. Otherwise, True.

Level

Page or Field level.

Details

Parent topic: [Validations actions](#)

FieldContainsValue

Determines whether the current field value contains some or all of the specified text but no additional text.

Syntax

```
bool FieldContainsValue (StrParam)
```

Parameters

Text that the action is looking for in the current field. Smart parameters are supported.

Returns

False if the *CurrentObj.Text* variable of the object does not contain the parameter value. Otherwise, True.

Level

All, but generally at the Field level.

Details

This action determines if a field represented by the bound object of the Document Hierarchy contains some or all of the parameter's text value, without additional unspecified text.

Example

```
FieldContainsValue("NEW")
Parameter: "NEW" = TRUE
Parameter: "NEW Action" = TRUE
Parameter: "Project" = FALSE
Parameter: "NEW Project" = TRUE
Parameter: "Development" = FALSE
```

Parent topic: [Validations actions](#)

FilterFieldSelectedChars

Deletes the specified characters from the value of the current field.

Syntax

```
bool FilterFieldSelectedChars (StrParam)
```

Parameters

A String containing the character or characters to be removed.

Every instance of the characters are removed from the captured value.

Returns

Always True.

Level

Field level.

Details

Removes all instances of the characters that you enter as parameters from the captured value of the current Field object.

Example

```
FilterFieldSelectedChars(0)
11002900 becomes 1129
```

Parent topic: [Validations actions](#)

Related reference:

[DeleteSelectedChars](#)

FormatNumberToLocale

Evaluates the current field value for known number patterns and if a known pattern is detected, updates the decimal and digit separators characters to match that of the current locale.

Syntax

```
bool FormatNumberToLocale ()
```

Returns

True, if no errors are encountered. Otherwise, False.

Level

Field level.

Details

This action evaluates the current field value for known number patterns, If a known pattern is detected, the action updates the decimal and digit separators characters (if present) to match the current locale. For use when processing fields with number types formatted incorrectly for the current locale, such as US versus European locales. Known patterns are industry standard digits with or without 3-digit group separators, and 1-2 digits following a decimal separator.

Distinct Analyzed groups:

1. Numerals with decimal point
 - Group Separators using comma, apostrophe, or space characters.
 - Decimal Separator using decimal character.
2. Numerals with decimal comma
 - Group Separator using decimal, apostrophe, or space characters.
 - Decimal Separator using comma character.
3. Numerals with Arabic/Persian characters
 - Group Separator using Arabic/Persian character.
 - Decimal Separator using Arabic/Persian character.

Note: Numerals with lakhs layout and decimal mark (pattern of `n, nn, nn, nnn.dd`) are not supported.

Example

a) Given a US number 1,234.56 to be formatted to UK.
`FormatNumberToLocale()`
New format will be 1.234,56

b) Given a US number 1234.5 to be formatted to UK.
`FormatNumberToLocale()`
New format will be 1234,5

c) Given a US number 1,234.567 to be formatted to UK.
`FormatNumberToLocale()`
Format will remain 1,234.567 since 3 digits following the decimal are not supported.

d) Given a US number 12345678.90 to be formatted to UK.
`FormatNumberToLocale()`
New format will be 12345678.90

e) Given a UK number 1.234.567,89 to be formatted to US.
`FormatNumberToLocale()`

New format will be 1,234,567.89

f) Given a UK number 0,5 to be formatted to US.
FormatNumberToLocale()
New format will be 0.5

Parent topic: [Validations actions](#)

GetJobID

Assigns the current job ID to the Text property of the current object.

Syntax

```
bool GetJobID ()
```

Parameters

None.

Returns

False if the action cannot find a JobID value. Otherwise, True.

Level

All.

Details

Assigns the Job ID of the current User Application job (the Pilot.JobID property of the job) to the *CurrentObj.Text* variable of the bound object of the Document Hierarchy.

Example

```
GetJobID ()
```

Parent topic: [Validations actions](#)

HasChildOfType

Determines whether the current object has a child of the specified type.

Syntax

```
bool HasChildOfType (StrParam)
```

Parameters

The name of a level of the Document Hierarchy (Batch, Document, Page, Field) or of a runtime variable.

Returns

False if the bound object does not include a child or children specified by the parameter, or a variable identified by the parameter. Otherwise, True.

Level

All.

Details

Determines if the bound object of the Document Hierarchy has a child or children of the type specified by the parameter. The action can also determine if a runtime variable specified as a parameter has been assigned to the bound object.

Example

```
HasChildOfType ("Page")
```

This example determines if the bound object is the parent of one or more pages.

```
HasChildOfType ("IGNORE")
```

In this example, the action determines if IGNORE is a runtime variable of the bound object.

Parent topic: [Validations actions](#)

InsertChars

Inserts one or more characters into the value of the current field.

Syntax

```
bool InsertChars (strParam)
```

Parameters

1. The characters or character string to be inserted; defaults to a space.
2. A number n indicating the target position within the captured value; defaults to the end of the value.
3. The number of insertions; defaults to 1.

Returns

Always True.

Level

Field level.

Details

Inserts a character or string of characters into the captured value, one or more times.

Example

```
InsertChars ("=$,1,1")  
345.67 becomes =$345.67
```

```
InsertChars ("=$,1,2")
345.67 becomes  =$=$345.67
```

Parent topic: [Validations actions](#)

InsertDecimalPoint

Inserts a decimal point into the value of the current field at the specified position.

Syntax

```
bool InsertDecimalPoint (strParam)
```

Parameters

A number *n* indicating the character position at which to place the decimal. Smart parameters are supported.

Returns

Always True.

Level

Field level.

Details

Places a decimal character in the captured value, at the character position specified as a parameter. The parameter indicates the position of the decimal point, moving from right to left.

Example

```
InsertDecimalPoint ("2")
324556 becomes 3245.56
```

```
InsertDecimalPoint ("2")
355 becomes 3.55
```

Parent topic: [Validations actions](#)

IsFieldCurrency

Determines if the captured value of the Field meets the currency format of the current locale.

Syntax

```
bool IsFieldCurrency ()
```

Parameters

None.

Returns

True if the current locale's format criteria are met. Otherwise, False.

Level

Field level.

Details

This determination includes the number of decimal places, decimal and digit separator characters and any present currency indicators.

Example

```
IsFieldCurrency()  
$1,200 returns False  
$1,200.00 returns True
```

Parent topic: [Validations actions](#)

IsFieldDate

Checks that the value of the field has an acceptable Date format. This action uses the current locale setting to determine valid patterns.

Syntax

```
bool IsFieldDate ()
```

Parameters

None.

Returns

True if the specifications of the action are met. Otherwise, False.

Level

Field level.

Details

This action accepts any valid date from January 1st year 1 through December 31st 9999.

Example

```
IsFieldDate()  
  
In locale en-US (United States):  
April 6, 1944 returns True  
04/06/44 returns True  
30.6.44 returns False  
Feb 31,2003 returns False
```

Parent topic: [Validations actions](#)

IsFieldDateEqualOrAfter

Checks that the Date value in the current field represented by the bound Field object of the Document Hierarchy is greater than or equal to the Date value in the field that is specified as the parameter.

Syntax

```
bool IsFieldDateEqualOrAfter (StrParam)
```

Parameters

The name of the Field object of the Document Hierarchy to be compared with the Date value of the current field.

Returns

False if the date condition is not met, if the action is not applied at the Field level, or either field does not contain a valid date. Otherwise, True.

Level

Field level.

Details

Example

```
IsFieldDateEqualOrAfter ("24aDtFr1")
```

Parent topic: [Validations actions](#)

Related reference:

[IsFieldDateEqualOrBefore](#)

IsFieldDateEqualOrBefore

Checks that the date in the current field represented by the bound Field object of the Document Hierarchy is less than or equal to the Date value in the field that is specified as the parameter.

Syntax

```
bool IsFieldDateEqualOrAfter (StrParam)
```

Parameters

The name of the Field object of the Document Hierarchy to be compared with Date value of the current field.

Returns

False if the date condition is not met, if the action is not applied at the Field level, or if either field does not contain a valid date value. Otherwise, True.

Level

Field Level.

Details

Example

```
IsFieldDateEqualOrBefore ("24aDtFr1")
```

Parent topic: [Validations actions](#)

Related reference:

[IsFieldDateEqualOrAfter](#)

IsFieldDateUpToToday

Checks that the Date value of the current Field object is today's date or earlier.

Syntax

```
bool IsFieldDateUpToToday ()
```

Parameters

None.

Returns

False if the value of the field is not a valid date or if the date is after Today. Otherwise, True.

Level

Field level.

Details

Example

```
IsFieldDate ()  
IsFieldDateUpToToday ()
```

This sequence confirms that a value is a date and that it is the same as, or earlier than, today's date.

Parent topic: [Validations actions](#)

Related reference:

[IsFieldDate](#)

[DateStampField](#)

IsFieldDateWithinRange

Checks that the value assigned to the Text property of the bound object is a valid Date. If yes, this action then confirms that the Date is within the range specified by the parameters.

Syntax

```
bool IsFieldDateWithinRange (StrParam)
```

Parameters

Comma-separated Dates that define the range:

1. Start Date
2. End Date

TODAY can represent the current Date. Smart parameters are supported.

Returns

False if the value assigned to the Text property of the current object is not a valid date; if either parameters is invalid; or the Date value of the Text property is not within the range that is specified by the parameters. Otherwise, True.

Level

All, but generally at the Field Level.

Details

Example

```
IsFieldDateWithinRange ("1/1/2006, 1/31/2006")  
IsFieldDateWithinRange ("1/1/2006, TODAY")
```

Parent topic: [Validations actions](#)

IsFieldDateWithinXDays

Checks that the captured Date value of the current Field object is within *n* days of the number entered as a parameter.

Syntax

```
bool IsFieldDateWithinXDays (StrParam)
```

Parameters

A number *n* that specifies how many days make up the review period. Smart parameters are supported.

Returns

False if the value of the field is not a valid date or if the date is older than the number of days in the parameter. Otherwise, True.

Level

Field level.

Details

Example

```
IsFieldDate()  
IsFieldDateWithinXDays("30")
```

This sequence checks that a value is a date within 30 days of today's date.

Parent topic: [Validations actions](#)

Related reference:

[IsFieldDate](#)

[IsFieldDateUpToToday](#)

IsFieldDateWithReformat

Confirms that the data of a field is a valid date and then formats the Date value according to the format entered as the parameter.

Syntax

```
bool IsFieldDateWithReformat (StrParam)
```

Parameters

The Date format you want to use.

- mm/dd/yyyy
- mm/dd/yy
- dd/mm/yy
- mm.dd.yy, etc.

Defaults to system short date format if no format or single * is used. Smart parameters are supported.

Returns

False if the parameter is invalid, or the current field value is not a valid date given the specified format. Otherwise True.

Level

Field level.

Details

Example

```
IsFieldDateWithReformat("*")  
June 3, 2002 becomes 06/03/2002  
  
IsFieldDateWithReformat("mm.dd.yy")  
June 3, 2002 becomes 06.03.02
```

Parent topic: [Validations actions](#)

IsFieldEmpty

Checks that the Field object designated as a parameter does not have a captured value.

Syntax

```
bool IsFieldEmpty (strParam)
```

Parameters

The name of the Field object.

Returns

False if the name of the Field object does not exist or if the field contains a captured value. Otherwise, True.

Level

All.

Details

Example

```
IsFieldEmpty("Shipping")  
AssignFieldDefault("NoShipping")
```

In this example, if the captured value of the Shipping Field object is \$921.11, this action return a False condition and terminates the rule.

If the Shipping Field object does not have a value (True), the AssignFieldDefault action enters the NoShipping parameter as the captured value of the Field object.

Parent topic: [Validations actions](#)

Related reference:

[AssignFieldDefault](#)

[IsFieldFilled](#)

IsFieldFilled

Determines whether the Field object designated as a parameter contains a captured value or is empty.

Syntax

```
bool IsFieldFilled (strParam)
```

Parameters

The name of the Field object.

Returns

False if the name of the Field object does not exist or if the field does not contain a captured value. Otherwise, True.

Level

All.

Details

Example

```
IsFieldFilled("PaymentDue")
```

If the action returns True because the field does contain a value, the rule invokes its next action.

If the action returns False, the rule closes and the task applies the next rule, which might include a CopyFieldToField action.

Parent topic: [Validations actions](#)

Related reference:

[AssignFieldDefault](#)

[IsFieldEmpty](#)

IsFieldGreaterOrEqual

Determines if the captured value of the current Field object is greater than or equal to the value entered as a parameter.

Syntax

```
bool IsFieldGreaterOrEqual (strParam)
```

Parameters

The Numeric or Currency value which is the basis for comparison. Smart parameters are supported.

Returns

False if the parameter or the captured value of the Field object is not Numeric. Or if the result does not meet the action's requirements. Otherwise, True.

Level

Field level.

Details

If the value of the field is not Numeric or currency, the action returns a False condition.

Example

```
IsFieldGreaterOrEqual("624")  
Returns True if the Field object's value is 625.00  
Returns False if the value is 623.99.  
Returns True if the value is 624.00.
```

Parent topic: [Validations actions](#)

Related reference:

IsFieldHidden

Checks that the calling field is Hidden, which means the field has a variable *STATUS* that is equal to -1.

Syntax

```
bool IsFieldHidden ()
```

Parameters

None.

Returns

True if the *STATUS* variable of the Field object = -1. Otherwise False.

Level

Field level.

Details

This action returns True if the calling field is Hidden, the corresponding Field object of the Document Hierarchy has a variable *STATUS* that is equal to -1.

Example

```
IsFieldHidden ()
```

Parent topic: [Validations actions](#)

IsFieldLengthMax

Checks that the character length of the current Field object's captured value is equal to or less than the value set as a parameter.

Syntax

```
bool IsFieldLengthMax (StrParam)
```

Parameters

A number *n* designating the maximum length of the value. Smart parameters are supported.

Returns

False if the parameter that you enter is not numeric, or if the number of characters exceeds the value of the parameter. Otherwise, True.

Level

Field level.

Details

Example

```
IsFieldLengthMax("6")  
EU2240 returns True  
EU002240 returns False
```

Parent topic: [Validations actions](#)

Related reference:

[IsFieldLengthMin](#)

IsFieldLengthMin

Checks the character length of the current Field object's captured value to see if its length is equal to or longer than a number *n*.

Syntax

```
bool IsFieldLengthMin (StrParam)
```

Parameters

A number *n* designating the minimum length of the value. Smart parameters are supported.

Returns

False if the parameter that you enter is not numeric, or if the number of characters is less than the parameter's value. Otherwise, True.

Level

Field level.

Details

Example

```
IsFieldLengthMin("6")  
EU2240 returns True  
EU22 returns False
```

Parent topic: [Validations actions](#)

Related reference:

[IsFieldLengthMax](#)

IsFieldLessOrEqual

Determines if the captured value of the current Field object is less than or equal to the value entered as a parameter.

Syntax

```
bool IsFieldLessOrEqual (strParam)
```

Parameters

The Numeric or Currency value you want to compare against. Smart parameters are supported.

Returns

False if the parameter or captured value of the Field object is not Numeric. Or if the result does not meet the action's requirements. Otherwise, True.

Level

Field level.

Details

If the value of the field is not Numeric or Currency, the action returns a False condition.

Example

```
IsFieldLessOrEqual("625")  
Returns True if the Field object's value is 624.99  
Returns False if the value is 625.01  
Returns True if the value is 625.00
```

Parent topic: [Validations actions](#)

Related reference:

[IsFieldGreaterOrEqual](#)

IsFieldMatching

Determines if the value entered as the parameter is identical to the captured value of the current Field object.

Syntax

```
bool IsFieldMatching (strParam)
```

Parameters

The value to be checked against the value of the Field object.

Returns

True if the requirement of the action is met. Otherwise, False.

Level

Field level.

Details

Example

If the Field object's value is 525.00:

```
IsFieldMatching("525.00") returns True  
IsFieldMatching("525") returns False
```

Parent topic: [Validations actions](#)

Related reference:

[IsFieldPercentAlpha](#)

[IsFieldPercentNumeric](#)

IsFieldPercentAlpha

Determines if the characters in the captured value of current Field object are *n%* alphabetic.

Syntax

```
bool IsFieldPercentAlpha (StrParam)
```

Parameters

A number (0-100) indicating the percentage necessary to return a True condition. The value must be numeric, without the percent sign. Smart parameters are supported.

Returns

True if the requirements of the parameter are met. Otherwise False, including if the Field is empty.

Level

Field level.

Details

Example

```
IsFieldPercentAlpha("50") #RPR-1421 returns False  
IsFieldPercentAlpha("30") #RPR1421 returns True
```

Parent topic: [Validations actions](#)

Related reference:

[IsFieldPercentNumeric](#)

[IsFieldMatching](#)

IsFieldPercentNonNumeric

Determines if any of the characters in the captured value of the current Field object are *n%* not numeric characters.

Syntax

```
bool IsFieldPercentNonNumeric (StrParam)
```

Parameters

A number (0-100) indicating the percentage that results in a True condition. The default percentage is 100. The value must be numeric, without the percent sign. Smart parameters are supported.

Returns

False if the parameter is non-numeric, if the field is empty, or if the value of the field exceeds the parameter's percentage of numeric characters. Otherwise, True.

Level

Field level.

Details

This determination includes and is not limited to valid decimal and numeric separator characters.

Example

```
Given the current value is "1,234.56US"  
(Percentage of non-numeric characters is 40%)  
IsFieldPercentNonNumeric("0") returns True  
IsFieldPercentNonNumeric("30") returns True  
IsFieldPercentNonNumeric("40") returns True  
IsFieldPercentNonNumeric("70") returns False  
IsFieldPercentNonNumeric("100") returns False
```

Parent topic: [Validations actions](#)

Related reference:

[IsFieldPercentAlpha](#)

[IsFieldMatching](#)

IsFieldPercentNumeric

Determines if the characters in the captured value of the current Field object are *n*% numeric characters.

Syntax

```
bool IsFieldPercentNumeric (StrParam)
```

Parameters

A number (0-100) indicating the percentage that results in a True condition. The default percentage is 100. The value must be numeric, without the percent sign. Smart parameters are supported.

Returns

False if the parameter is non-numeric, if the field is empty, or if the value of the field exceeds the parameter's percentage of numeric characters. Otherwise, True.

Level

Field level.

Details

This determination does not include, and is not limited to, valid decimal and numeric separator characters.

Example

```
Given the current value is "1,234.56US"  
(Percentage of numeric characters is 60%)  
IsFieldPercentNumeric("0") returns True  
IsFieldPercentNumeric("50") returns True  
IsFieldPercentNumeric("60") returns True  
IsFieldPercentNumeric("70") returns False  
IsFieldPercentNumeric("100") returns False
```

Parent topic: [Validations actions](#)

Related reference:

[IsFieldPercentAlpha](#)

[IsFieldMatching](#)

IsMatchingJobID

Checks that the Job ID of the current User Application job matches the Job ID value of the parameter.

Syntax

```
bool IsMatchingJobID (StrParam)
```

Parameters

String value of the Job ID to be compared to the current Job ID.

Returns

True if the current Job ID matches the value of the parameter. Otherwise, False.

Level

All.

Details

Example

```
IsMatchingJobID("Main")
```

Parent topic: [Validations actions](#)

IsMaxOMRChecked

Indicates the maximum number of check boxes that can contain a value, such as a check.

Syntax

```
bool IsMaxOMRChecked (StrParam)
```

Parameters

An Integer value specifying this maximum. Smart parameters are supported.

Returns

False if the parameter you enter is not numeric, or the field is not an OMR field.

True if the number of OMR boxes checked is less than or equal to the parameter value that you entered.

Level

Field level.

Details

Example

```
IsMaxOMRChecked("1")
```

Parent topic: [Validations actions](#)

Related reference:

[IsMinOMRChecked](#)

IsMinOMRChecked

Indicates the minimum number of check boxes that can contain a value, such as a check.

Syntax

```
bool IsMinOMRChecked (StrParam)
```

Parameters

An Integer value specifying this minimum. Smart parameters are supported.

Returns

False if the parameter you enter is not numeric, or the field is not an OMR field.

True if the number of OMR boxes checked is greater than or equal to the parameter value that you entered.

Level

Field level.

Details

Example

```
IsMinOMRChecked("1")
```

Parent topic: [Validations actions](#)

Related reference:

IsPatternInField

Checks that the value of the current field contains the specified VBScript regular expression.

Syntax

```
bool IsPatternInField (StrParam)
```

Parameters

String value of the Regular Expression. The expression can include any Regular Expression characters. Smart parameters are supported.

Returns

True, if the pattern is found within the field. Otherwise, False.

Level

All, but generally at the Field level.

Details

Uses VBScript Regular Expression Pattern entered as parameter to search for a matching pattern in the Text value of the current object.

Attention: If you are using an input boundary, ^, it must be followed by a space and then the remainder of the search string.

Example

```
IsPatternInField("[\^\b\s\n\r]*Inv[oO][iItl1]ce[\b\s]*")  
IsPatternInField("@STRING([\^\b\s\n\r]*Inv[oO][iItl1]ce[\b\s]*")
```

This example searches for the word “Invoice” in the current field. To allow for recognition errors, the search allows for common recognition substitutions in “o” and “i” by matching “Invoice”, “InvOice”, “InvOice”, “Inv01ce”, “Involce”, and so on. The search also ignores any text that is placed before or after the word. Encapsulating the parameter with @STRING is recommended when the value is not a Smart Parameter.

Parent topic: [Validations actions](#)

IsSupportedImageFile

Checks that the image file attached to the current page is in a supported image format.

Syntax

```
bool IsSupportedImageFile (StrParamInput)
```

Parameters

A Boolean that determines the type of test to perform.

True: Tests the validity of the image by checking the file extension and attempting to load the image.

False: Only a file extension check is performed to determine if it is a supported file.

Returns

True if parameter is valid, the action is called at the page level and the page's *IMAGEFILE* variable (set at scan time by the scan tasks) points to a file whose format is supported (can be displayed) by the Image view control. Otherwise False.

Level

Page level.

Details

This action tests a file to determine if the file format is supported. The extension is checked to determine if it denotes a supported image type. If True is passed as a parameter, it also attempts to load the file into the Image view control.

Using False as a parameter improves the speed of this action. A parameter of True slows down the processing, especially if the images are very large, but it adds an extra confirmation that the file is not corrupted and confirm that any subformat of the file type, such as compression type, is also supported.

Example

```
IsSupportedImageFile ()
```

Parent topic: [Validations actions](#)

IsThisFieldEmpty

Checks that the value of the current field is empty.

Syntax

```
bool IsThisFieldEmpty ()
```

Parameters

None.

Returns

False if not applied to the Field level, or if the current field has a text value. Otherwise, True.

Level

Field level.

Details

Confirms if the current field has no captured value.

Example

```
IsThisFieldEmpty()
```

Parent topic: [Validations actions](#)

Related reference:

[IsThisFieldFilled](#)

[IsFieldEmpty](#)

IsThisFieldFilled

Checks that the current field has a captured value.

Syntax

```
bool IsThisFieldFilled ()
```

Parameters

None.

Returns

False if not applied to the Field level, or if the current field has no text value. Otherwise, True.

Level

Field level.

Details

Confirms if the current field has a captured value.

Example

```
IsThisFieldFilled()
```

Parent topic: [Validations actions](#)

Related reference:

[IsThisFieldEmpty](#)

[IsFieldFilled](#)

IsVariableEmpty

Checks that the variable specified by the parameter does not contain a value.

Syntax

```
bool IsVariableEmpty (strParam)
```

Parameters

Name of the variable of the current object to be checked.

Returns

False if the parameter is invalid, or if the variable contains a value. Otherwise, True.

Level

All.

Details

This action only checks variables of the current object.

Example

```
IsVariableEmpty("TemplateID")
```

Parent topic: [Validations actions](#)

Related reference:

[IsVariableFilled](#)

IsVariableFilled

Checks that the variable specified by the parameter contains a value.

Syntax

```
bool IsVariableFilled (strParam)
```

Parameters

Name of the variable of the current object to be checked.

Returns

False if the parameter is invalid, or if the variable does not contain a value. Otherwise, True.

Level

All.

Details

This action only checks variables of the current object.

Example

```
IsVariableFilled("TemplateID")
```

Parent topic: [Validations actions](#)

Related reference:

[IsVariableEmpty](#)

ParseMultilineAddress

Splits the value of the current field at each comma and saves the substrings to the specified fields. Typically used for address fields.

Syntax

```
bool ParseMultilineAddress (strParam)
```

Parameters

Comma-separated Smart Parameter String values of the names of fields that hold the parsed data, in the following order:

Name, AddressLine1, AddressLine2, City, State, Zip code or postal code, Phone

Returns

False, if the parameters are invalid or parsing cannot take place. Otherwise, True.

Level

Field level.

Details

Parses a multiline US address Field object's captured value.

Comma-separated String values of the names of fields that hold the parsed data, in the following order: Name, AddressLine1, AddressLine2, City, State, Zip code or postal code, and Phone. The example assumes that fields are sibling fields of the calling object. For other relationships, review smart parameter syntax.

Expected Pattern:

- Phone Number (optional)
- Name
- Address Line One
- Address Line Two (optional)
- City, State, Zip code or postal code
- Phone Number (optional)

Note: Parsing logic assumes that State and Zip or postal code are on the same address line. Only one Phone number per address field is supported. The expected pattern shows the two optional positions for this value.

Example

```
ParseMultilineAddress ("VendorName,VenAddress1,VenAddress2,VenCity,VenState,VenZip,VenPhone")
```

Parent topic: [Validations actions](#)

ParseName

Splits the three word value of the current field and saves the substrings to the specified fields. Typically used for name fields, such as first name, middle name/initial, last name.

Syntax

```
bool ParseName (StrParam)
```

Parameters

Three comma separated parameters:

1. The name of the Last Name Field object.
2. The name of the First Name Field object.
3. The name of the Middle Name or Middle Initial Field object.

Returns

False if not placed at the Field level; if the current field contains no data; or if the parameters are invalid. Otherwise, True.

Level

Field level.

Details

Parses the captured values of a name Field object. Applied to a Name field, the action parses the Last, First, and Middle names into the fields specified by the parameter.

Example

```
ParseName ("LastName, FirstName, MidName")
```

Bound to a Name Field object which includes values for all three names, the action place the Last name into the LastName field, the First name into the FirstName field, and the Middle name (or middle initial) into the MiddleName field.

Parent topic: [Validations actions](#)

Related reference:

[ParseMultilineAddress](#)

ReplaceChars

Replaces a character or string of characters in the captured value of the current Field object with a String that you enter as one of the parameters.

Syntax

```
bool ReplaceChars (StrParam)
```

Parameters

1. The character or string of characters to be replaced; defaults to a space. Smart Parameter Enabled.
2. The characters of the replacement String. Smart Parameter Enabled.

3. The number of times replacement is to occur. The default is 1 and * replaces all instances.

Returns

Always True.

Level

Field level.

Details

Example

```
ReplaceChars(".", "/", "*")  
01.02.2005 becomes 01/02/2005
```

Parent topic: [Validations actions](#)

ReplaceValueAtPosition

Replaces the value at the specified position within the current field with a replacement string, or deletes the value.

Syntax

```
bool ReplaceValueAtPosition (StrParam)
```

Parameters

Two-part comma-separated value:

1. The position that contains the value to be replaced.
2. The replacement string; this parameter defaults to "" indicating a deletion.

Returns

True if the character replacement is successful. Otherwise, False.

Level

Field level.

Details

Example

```
ReplaceValueAtPosition("3, /")
```

Parent topic: [Validations actions](#)

ResetField

Deletes the value of the current field and sets the Position attribute of the field to 0,0,0,0.

Syntax

```
bool ResetField ()
```

Parameters

None.

Returns

False if not called on the field level. Otherwise, True.

Level

Field level.

Details

The action does not clear any Alt-Text values that are associated with the bound Field object. That is the responsibility of a ClearAltText action.

Example

```
ResetField()
```

This action typically belongs to a follow-up Validate rule that deals with a False response to an action such as IsDate() in the previous rule. If the value of the field is not a date (in this example), the ResetField action removes the value that is there.

This action example also sets the Position attribute of the field to 0,0,0,0.

Parent topic: [Validations actions](#)

SetIsOverrideable

Specifies if the user can or cannot override a validation that fails for the current object.

Syntax

```
bool SetIsOverrideable (StrParam)
```

Parameters

None.

Returns

Always True.

Level

All.

Details

Important: This status might prevent an operator from overriding validations on a field and then continuing to subsequent pages.

Example

```
SetIsOverrideable("False")  
IsFieldPercentNumeric("100")
```

In this sequence, if the captured value of the field is not 100% Numeric, an operator cannot override a Validation rule's subsequent rejection of the value.

Parent topic: [Validations actions](#)

SplitFieldValuePreserveEnd

Splits the captured value of a Field object at the first instance of the character specified as a parameter.

Syntax

```
bool SplitFieldValuePreserveEnd (strParam)
```

Parameters

String value of the separating character. Smart parameters are supported.

Returns

True if the separator character is found. Otherwise False.

Level

Field level.

Details

The action deletes all characters prior to the separating character, as well as the separation character. All text after the separating character remains.

Example

```
SplitFieldValuePreserveEnd("=")
```

If the value of the object is InvNumber=A1234, this action truncates it to A1234.

Parent topic: [Validations actions](#)

Related reference:

[SplitFieldValuePreserveStart](#)

SplitFieldValuePreserveStart

Splits the captured value of a Field object at the first instance of the character specified as a parameter.

Syntax

```
bool SplitFieldValuePreserveStart (strParam)
```

Parameters

String value of the separating character. Smart parameters are supported.

Returns

True if the separator character is found. Otherwise False.

Level

SplitFieldValuePreserveEndField level.

Details

The action deletes all characters to the end of the string starting from the separating character, as well as the separation character. All text prior to the separating character remains.

Example

```
SplitFieldValuePreserveStart("c")
```

If the value of the object is Description, this action truncates it to Des.

Parent topic: [Validations actions](#)

Related reference:

[SplitFieldValuePreserveEnd](#)

SumFields

Adds the values of all child fields of the specified type and assigns the result to the current field. You can also use this action to sum the values of the specified variable for all child objects.

Syntax

```
bool SumFields (StrParam)
```

Parameters

String value of a Field object's Type property or the name of a variable.

Returns

Always True.

Level

All.

Details

Sums captured values of any child Field if a child object's Type property is identical to the Type you specify as a parameter. Alternatively, the actions sums values assigned to a variable of the child Field objects. In this case, the variable is the same as the variable entered as a parameter.

Remember: This action must be applied to the parent object.

Example

```
SumFields("Detail")
```

```
SumFields("LineTotal")
```

The first action in the example sums the captured values of Detail Field objects that are children of the current Field object.

The second action in the example sums values assigned to the *LineTotal* variable of child Field objects.

In both cases, the result is a Long number.

Parent topic: [Validations actions](#)

TimeStampField

Updates the current Field object with the current time.

Syntax

```
bool TimeStampField (StrParam)
```

Parameters

A time format, for example, HH:MM:SS, or HH:MM.

* defaults to HH:MM:SS. Smart parameters are supported.

Returns

Always True.

Level

Field level.

Details

Example

```
TimeStampField("*") produces  
09:20:02
```

```
TimeStampField("HH:MM") produces  
09:20
```

Parent topic: [Validations actions](#)

Related reference:

TrimSpaces

Deletes extra spaces at the beginning and end of the captured value of the current Field object.

Syntax

```
bool TrimSpaces ()
```

Parameters

None.

Returns

Always RightTruncate.

Level

Field level.

Details

Example

```
TrimSpaces ()  
456.11_ _ _ _ becomes 456.11
```

Parent topic: [Validations actions](#)

Related reference:

[RightTruncate \(deprecated\)](#)

TruncateFromEnd

Deletes characters from the end of the captured value of the current Field object until the length of the value equals the length indicated by the parameter.

Syntax

```
bool TruncateFromEnd (StrParam)
```

Parameters

A Number n that is the maximum length of the value. Smart parameters are supported.

Returns

False if the parameter is not Numeric. Otherwise, True.

Level

Field level.

Details

This action deletes characters from the end of the captured value of the current Field object until the length of the value equals the length indicated by the parameter.

Example

```
TruncateFromEnd("6")  
EU0002240 becomes EU0002
```

Parent topic: [Validations actions](#)

Related reference:

[TruncateFromStart](#)

[IsFieldLengthMax](#)

TruncateFromStart

Deletes characters from the start of the captured value of the current Field object until the length of the value equals the length specified by the parameter.

Syntax

```
bool TruncateFromStart (StrParam)
```

Parameters

A number n that is the maximum length of the value. Smart parameters are supported.

Returns

Always True.

Level

Field level.

Details

This action deletes characters from the start of the captured value of the current Field object until the length of the value equals the length specified by the parameter.

Example

```
TruncateFromStart("6") reduces the following value:  
3,344.01 becomes 344.01
```

Parent topic: [Validations actions](#)

Related reference:

[TruncateFromEnd](#)

Vote actions

Use the Vote action when you do multi-pass data entry to check whether the first and second passes match.

The Vote action returns True if the data entered for by the first operator matches the data that is entered by the second operator.

- [VoteFld](#)
Checks to see whether the data entered by the first Data Entry operator matches the data that is entered by the second Data Entry operator.

Parent topic: [Global actions](#)

VoteFld

Checks to see whether the data entered by the first Data Entry operator matches the data that is entered by the second Data Entry operator.

Syntax

```
bool VoteFld ()
```

Parameters

None.

Returns

False, if the values do not match. Otherwise, True.

Level

Field level.

Details

Checks to see whether the data entered by the first Data Entry operator matches the data entered by the second Data Entry operator.

Failed (mismatched) values set the confidence for the entire string to '1', which flags the field as Low Confidence. Positive matches set the entire strings confidence to '9' (High Confidence).

This action needs to run after the second Data Entry task is complete.

This action is used for workflows by using third pass data entry.

Example

```
VoteFld()
```

Parent topic: [Vote actions](#)

Vscan actions

Use the Vscan actions to create a batch by using existing image files.

Important: If any concurrently running threads might be using the same source directory, your Vscan-related tasks cannot be run in a multi-threaded configuration. Instead, use single-threaded tasks only. For information about configuring threads in Rulerunner, see [Rulerunner thread configuration](#). For information about the source directory, see [SetSourceDirectory](#).

The Vscan actions determine which documents are included in the batch and how to handle these documents as part of the scan task.

- [AddDocument](#)
Adds a document node to the runtime hierarchy. All scanned pages become children of the document node.
- [CopyFile](#)
When used before the Scan action, this action tells the Scan action to also copy the files to the specified location.
- [DeleteImageFile](#)
When used before the Scan action, this action tells the Scan action to delete the source files from the images folder.
- [MoveImageFileToDirectory](#)
When used before the Scan action, this action tells the Scan action to move the files from the images folder to the specified location.
- [Scan](#)
Copies image files from the location that is specified by the SetSourceDirectory action to the batch folder and creates the runtime hierarchy.
- [SearchInSubdirectory](#)
When used before the Scan action, this action tells the Scan action to look in subdirectories of the images folder.
- [SetAlternateImageNames](#)
Sets the file naming scheme of the Scan action.
- [SetFastMode](#)
Increases speed by preventing the Scan action from opening the files it is scanning. This action is provided only for compatibility with older applications, it is no longer used.
- [SetImageType](#)
Specifies the extensions of image file types to use, defaults to .tif.
- [SetMaxImageFiles](#)
Limits the number of files the Scan action copies.
- [SetMultiPageTiff](#)
When used before the Scan action, this action tells the Scan action to split multipage TIFF files into single page TIFF files.
- [SetSortOrder](#)
Specifies the order in which image files are imported.
- [SetSourceDirectory](#)
Specifies the path to the images folder. This action must precede the Scan action.

Parent topic: [Global actions](#)

AddDocument

Adds a document node to the runtime hierarchy. All scanned pages become children of the document node.

Syntax

```
bool AddDocument ()
```

Parameters

None.

Returns

False if this action is invoked or called at the wrong level or if the setup DCO does not contain any document structure. In this case, the action will not add a document. Otherwise, True.

Level

Batch level.

Details

Places all scanned pages into a single, batch-wide document. This action must be called before the Scan action.

The action also adds ED (Expected Documents), AD (Actual Documents), EP (Expected Pages) and AP (Actual Pages) properties to the Page file (.xml) of a task.

This action allows a VScan task to create a batch that has all the same characteristics of a batch that is created using a standard scan task. It is not required to call this action. Use this action if your application relies on a document structure on import.

This action must precede the Scan action. If AddDocument is used and the Scan action is never subsequently called, the action will not be performed.

Example

```
AddDocument ()
SetSortOrder ("Name,ASC")
SetMaxImageFiles ("100")
Scan ()
```

Parent topic: [Vscan actions](#)

CopyFile

When used before the Scan action, this action tells the Scan action to also copy the files to the specified location.

Syntax

```
bool CopyFile (StrParamSP)
```

Parameters

1. The String value of the name of the target file system folder.
2. Optional. The file extension you wish to use for each file. If you provide an extension, it must not be preceded by a period. The action defaults to .tif if this parameter is blank.

Smart parameters are supported.

Returns

Always True. If the target directory does not exist or if the files cannot be created, the files are not copied but the action still returns True.

Level

Any level but usually the batch level.

Details

A housekeeping action that copies the current source image file to a location that you specify and specifies the copied file's extension.

The source image file will remain in the input directory while a copy is placed into the specified directory. This action is typically used when it is desired to place images into a specific location for archiving. A copy of the image will still be placed into the batch directory as well.

If the target directory already contains a file of the same name, it will give a unique name to the new file.

This action must precede the Scan action. The CopyFile sets an indicator for the Scan action to copy the file. If CopyFile is used and the Scan action is never subsequently called, the copy will not be performed. The destination directory must already exist or a message will be logged and no files will be copied.

Example

```
CopyFile("C:\ParentDirectory\Application\Images\copies,tif")
Scan()
```

This example copies the source image file to the copies folder and then adds it to the batch. It does not delete the image from the source folder.

```
CopyFile("@APPPATH(vscancopydir),tif")
Scan()
```

This variant example copies the source image file to the copies folder that was specified in the application service.

Parent topic: [Vscan actions](#)

Related reference:

[Scan](#)

DeleteImageFile

When used before the Scan action, this action tells the Scan action to delete the source files from the images folder.

Syntax

```
bool DeleteImageFile ()
```

Parameters

None.

Returns

Always True.

Level

Any level but usually the batch level.

Details

A housekeeping action that deletes the current source image file. A Scan action must follow this action, which will perform the delete during the vscan process. If Scan is never called, the files are not deleted.

Example

```
DeleteImageFile ()  
Scan ()
```

This sequence deletes the image files from their current location after they were added to the current batch folder.

Parent topic: [Vscan actions](#)

MoveImageFileToDirectory

When used before the Scan action, this action tells the Scan action to move the files from the images folder to the specified location.

Syntax

```
bool MoveImageFileToDirectory (SmartParam)
```

Parameters

The full path to the target location of the Image file. Smart parameters are supported.

Returns

False if the parameter is not a valid directory, or if permission to access/write to the directory is denied. Otherwise, True.

If the file cannot be written to the destination directory, the batch status will be set to abort.

Level

Any level but usually the batch level.

Details

A housekeeping action that moves the current Image file to a location you specify. The source image file will be removed from the input directory and moved to the specified directory. This action is typically used when it is desired to place images into a specific location for archiving. A copy of the image will still be placed into the batch directory.

If the target directory already contains a file of the same name, it will give a unique name to the new file.

This action must precede the Scan action. If MoveImageFileToDirectory is used and the Scan action is never subsequently called, the file move will not be performed. If the move to the new directory fails, the batch will be set to abort.

Example

```
MoveImageFiletoDirectory("C:\ParentDirectory\Application\backup")
SetMaxImageFiles("100")
Scan()
```

This sequence copies the source image files to the current batch, then moves the Image files to the specified directory.

```
MoveImageFiletoDirectory("@APPPATH(vscanmovedir)")
Scan()
```

This variant example uses a smart parameter to obtain the directory path from the application service.

Parent topic: [Vscan actions](#)

Scan

Copies image files from the location that is specified by the SetSourceDirectory action to the batch folder and creates the runtime hierarchy.

Syntax

```
bool Scan ()
```

Parameters

None.

Returns

False if a SetSourceDirectory action does not precede this action. False is also be returned if fast mode is enabled and SetMultiPageTiff was called. Otherwise, True.

Level

Batch level.

Details

Scans a set of waiting Image files, according to the parameters set by earlier actions.

This is usually the last action in a vScan rule.

Example

```
SetImageType(".tif")
SetSourceDirectory("@APPPATH(vscanimagedir)")
SetMaxImageFiles("100")
Scan()
```

These are the elements of a sample vScan Rule. The Scan action will load the specified number of images (if available) from the specified folder into the current batch folder.

Parent topic: [Vscan actions](#)

SearchInSubdirectory

When used before the Scan action, this action tells the Scan action to look in subdirectories of the images folder.

Syntax

```
bool SearchInSubdirectory ()
```

Parameters

None.

Returns

Always True.

Level

Any level but usually the batch level.

Details

Looks for source files in sub-directories of the directory you designated with a SetSourceDirectory action. For this action to take effect, it must be called before the Scan action.

Example

```
SetSourceDirectory ("@APPPATH (vscanimagedir) ")  
SearchInSubdirectory ()
```

In this example, if the scan directory obtained from the application service includes sub-directories, this action directs the Scan action to process the contents of the sub-directories.

Parent topic: [Vscan actions](#)

Related reference:

[SetSourceDirectory](#)

SetAlternateImageNames

Sets the file naming scheme of the Scan action.

Syntax

```
()
```

Parameters

0: Sets the naming scheme of the input files to the format used by eDocument Conversion actions.

Any other input value uses the traditional naming scheme of TM000001, TM000002, etc.

Returns

Always True.

Level

Any level but usually the batch level.

Details

If you are using the Convert action library, it is recommended that you enable this batch file naming scheme.

If this action is not called prior to the Scan action, then the Scan uses the traditional naming scheme of TM000001, TM000002, etc.

Example

```
SetAlternateImageNames ("0")  
Scan ()
```

Parent topic: [Vscan actions](#)

Related reference:

[Scan](#)

SetFastMode

Increases speed by preventing the Scan action from opening the files it is scanning. This action is provided only for compatibility with older applications, it is no longer used.

Syntax

```
bool SetFastMode ()
```

Parameters

None.

Returns

Always True.

Level

Any level but usually the batch level.

Details

This action prevents the testing that the TIF files are valid and avoid conversion of TIF files to G4 compression for processing. Calling this action will allow TIF files that are not compatible with Datacap to be placed into a batch.

This action is provided as backwards compatibility to older applications where this action was required to properly enable fast mode for PDF files.

SetImageType will automatically enable fast mode, if it is called with a extension other than TIF or JPG. For this action to take effect, it must be called before the Scan action.

Example

```
SetImageType (.pdf)
SetFastMode ()
Scan ()
```

Parent topic: [Vscan actions](#)

SetImageType

Specifies the extensions of image file types to use, defaults to .tif.

Syntax

```
bool SetImageType (StrParam)
```

Parameters

String value of the file's identifying extension. If you are listing multiple types, separate each one with a comma.

It is not necessary to include a period before each extension. The parameters are not case-sensitive.

Returns

True, if the parameter is one of the values that is specified above. Otherwise, False.

Level

Any level but usually the batch level.

Details

Uses the value of a file extension to specify the type of files the task scans.

This is an optional vScan action. The task scans .tif files by default. For this action to take effect, it must be called before the Scan action.

This action automatically enables Fast Mode scanning, if the input parameter is not a single extension of type .tiff, .tif, .jpeg, or.jpg.

Example

```
SetImageType (.tif)
Scan ()
```

This sequence scans only TIF files and does not enable Fast Mode.

```
SetImageType ("tiff")
Scan ()
```

This sequence scans only TIFF files and does not enable Fast Mode.

```
SetImageType ("bmp, jpg, jpeg, msg, tif, tiff, pdf, zip, doc, docx, xls, xlsx, eml, gif")  
Scan()
```

This sequence scans all file types listed and enables Fast Mode.

Parent topic: [Vscan actions](#)

SetMaxImageFiles

Limits the number of files the Scan action copies.

Syntax

```
bool SetMaxImageFiles (StrParam)
```

Parameters

An optional string value specifying the maximum number of files.

If you do not enter a value, the task will scan all images in the target folder (up to 32767 files) and this action returns True.

Returns

False if the parameter is not Numeric. Otherwise, True.

Level

Any level but usually the batch level.

Details

Limits the number of Image files the vScan task will add to a single batch. This action must be placed before the Scan action for the setting to take effect during Scan.

Example

```
SetMaxImageFiles ("50")  
Scan()
```

Sets the maximum number of files to add to a batch at fifty.

Remember: A vScan task is a Batch Creation task: it sets up a new batch each time it scans Image files.

Parent topic: [Vscan actions](#)

SetMultiPageTiff

When used before the Scan action, this action tells the Scan action to split multipage TIFF files into single page TIFF files.

Syntax

```
bool SetMultiPageTiff ()
```

Parameters

None.

Returns

Always True.

Level

Any level but usually the batch level.

Details

Permits the use of multipage source image files. For this action to take effect, it must be called before the Scan action.

This action cannot be used if fast mode is enabled. It also requires that SetImageType be called with only one extension of TIF, TIFF, JPG or JPEG. If this action is used when fast mode is enabled, the Scan action returns False.

Example

```
SetImageType(".tif")
SetMultiPageTiff()
Scan()
```

If the Scan action in this sequence encounters a multipage .tif file, then it reads each one into the current batch as a separate image, thereby bursting the multipage file into individual images.

Parent topic: [Vscan actions](#)

SetSortOrder

Specifies the order in which image files are imported.

Syntax

```
bool SetSortOrder (strParam)
```

Parameters

Two comma-separated String values:

Parameter 1. Designation of the images' sorting field can be specified with text or numerically:

- 1 or Name : The input file name.
- 2 or Type : The file type.
- 3 or DateCreated : The File creation date.
- 4 or DateLastAccessed : The last file access date.
- 5 or DateLastModified : The last file modification date.
- 6 or Size : The file size.

Parameter 2. Optional: ASC or 1 (Ascending), DESC or 2 (Descending). If you do not include this parameter, the action defaults to ASC (1).

Returns

False if the parameters are not valid. Otherwise, True.

Level

Any level but usually the batch level.

Details

Sets the order in which Image files will be imported to the batch. The input files can be sorted by their file name, file type, date created, date accessed, date modified or file size. Using the optional second parameter, you can control if the values are sorted in ascending or descending order.

For this action to take effect, it must be called before the Scan action.

Example

```
SetSortOrder ("Name, ASC")
Scan ()
```

Parent topic: [Vscan actions](#)

SetSourceDirectory

Specifies the path to the images folder. This action must precede the Scan action.

Syntax

```
bool SetSourceDirectory (SmartParam)
```

Parameters

String value of the directory's name and path. Instead, you can use a Smart Parameter, such as @APPPATH, to establish the name and path of the Source Directory.

Returns

False, if the parameter is blank or the directory does not exist. Otherwise, True.

If the source directory does not exist, the batch status is set to abort.

Level

Any level but usually the batch level.

Details

This action indicates the name and path of the directory that contains the Image files to be scanned.

This is a required action. A task that employs a vScan rule cannot process images unless it has this key locator. This action must be called before the scan action.

Example

In the following example, the full path to the images directory is coded directly into the application rules.

```
SetSourceDirectory("c:\ParentDirectory\Application\Images")
```

In the following example, @APPPATH obtains the vscan directory from the application service. This makes the application more flexible because the same application that is installed in two different environments, such as test and production, can use two different input directories. The subdirectory Input is appended to the path.

```
SetSourceDirectory("@APPPATH(vscanimagedir)+\+Input")
```

Parent topic: [Vscan actions](#)

Web Services actions

This library of actions facilitates communication with external web services. You can use these actions to retrieve formatted responses or image files. The actions can handle both XML and JSON formatted responses, with XML as the default choice. You will need to know the web service's endpoint, any applicable parameters, and the response type before using these actions.

The Web Services library includes actions to set the url, execute connection requests, download and upload files, and set header and parameter values.

- [WsClearHeaders](#)
Clear all the previously set web service headers.
- [WsClearParameters](#)
Clear all of the previously set web service parameters. Call this action if you need to send a new set of parameters for your next call to any web service.
- [WsClearResultItems](#)
Clear all key-value pairs associating the result keys and previously set target DCO variables. Call this action if you need to send a new set of result items for your next call to any web service.
- [WsEncodeParameter](#)
Set a parameter to a specific encoding as required by the web service endpoint.
- [WsGetFile](#)
Downloads a file from a web service using the provided file name.
- [WsGetValues](#)
The WsGetValues action returns one or more values from a web service.
- [WsSetCredentials](#)
Associates credentials that you send with web service calls for authentication.
- [WsSetHeader](#)
Specify any header information about the response content.
- [WsSetNamespace](#)
Specify the namespace associated with the web service response.
- [WsSetParameter](#)
Set the parameters required by the web service endpoint.
- [WsSetResultItem](#)
This action will associate the keys(xpath or tags) to the values from the response to the DCO objects to which you want to assign them. If the source already exists its corresponding target will be replaced with the new target value.

- [WsSetTimeout](#)
The timeout value specified with this action will be used for all subsequent requests to web services.
- [WsUploadData](#)
Upload data to a Webservice endpoint using a POST request.
- [WsUploadFile](#)
Uploads a file to a web service.

Parent topic: [Global actions](#)

WsClearHeaders

Clear all the previously set web service headers.

Syntax

```
bool WsClearHeaders ()
```

Parameters

None.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Details

Call this action if you need to send a new set of headers for your next call to any web service.

Example

```
WsClearHeaders ()
```

Parent topic: [Web Services actions](#)

WsClearParameters

Clear all of the previously set web service parameters. Call this action if you need to send a new set of parameters for your next call to any web service.

Syntax

```
bool WsClearParameters ()
```

Parameters

None.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Example

```
WsSetParameter("FromCurrency", "USD", false)
WsSetParameter("ToCurrency", @APPVAR(values/gen/ToCurrency), false)
WsSetResultItem("double", "@B.ConversionRate")
WsGetvalues(@APPVAR(values/gen/CurrencyConvertor), "", "", "", "", "")
WsClearParameters()
WsClearResultItems()
WsSetParameter("address", "10 Downing Street, London", false)
WsSetParameter("sensor", "false", false)
WsSetResultItem("GeocodeResponse/result/geometry/location/lat",
"@B.Latitude")
WsSetResultItem("GeocodeResponse/result/geometry/location/lng",
"@B.Longitude")
WsGetvalues("http://maps.googleapis.com/maps/api/geocode/xml?", "", "", "",
"", "")
```

Parent topic: [Web Services actions](#)

WsClearResultItems

Clear all key-value pairs associating the result keys and previously set target DCO variables. Call this action if you need to send a new set of result items for your next call to any web service.

Syntax

```
bool WsClearResultItems ()
```

Parameters

None.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Example

```
WsSetParameter("FromCurrency", "USD", false)
WsSetParameter("ToCurrency", "GBP", false)
WsSetResultItem("double", "@B.ConversionRate")
```

```
WsGetvalues("http://www.websvcicex.net/CurrencyConvertor.asmx/ConversionRate?", "",
"", "", "", "")
    WsClearParameters()
    WsClearResultItems()
    WsSetParameter("address", "10 Downing Street, London", false)
    WsSetParameter("sensor", "false", false)
    WsSetResultItem("GeocodeResponse/result/geometry/location/lat",
"@B.Latitude")
    WsSetResultItem("GeocodeResponse/result/geometry/location/lng",
"@B.Longitude")
    WsGetvalues("http://maps.googleapis.com/maps/api/geocode/xml?", "", "", "",
"", "")
```

Parent topic: [Web Services actions](#)

WsEncodeParameter

Set a parameter to a specific encoding as required by the web service endpoint.

Syntax

```
bool WsEncodeParameter (string name, string encoding)
```

Parameters

name

The name of the parameter to be encoded.

encoding

The type of encoding to apply. URI or DATA types are supported.

Returns:

Always **True**.

Level

All levels.

Details

Call this action if you need to encode a parameter value for a call to any web service.

Example

```
WsSetParameter("CountryName", "United States", false)

WsSetParameter("CityName", "Newark", false)

WsEncodeParameter("CountryName", "data")

WsGetvalues("http://www.websvcicex.net/globalweather.asmx/GetWeather?", "",
"", "", "@B.htmltext", "")
```

Parent topic: [Web Services actions](#)

WsGetFile

Downloads a file from a web service using the provided file name.

Syntax

```
bool WsGetFile (string url, string filename, int timeout)
```

Parameters

url

Required: the URL of the web service. Smart parameters are supported.

filename

Required: the name to use for the downloaded file. Smart parameters are supported.

timeout

Optional: the timeout value in milliseconds before abandoning the request.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Details

Smart parameters are supported for all parameters. The timeout value specified for this operation will override any timeout value set elsewhere. The default timeout value is 60000 milliseconds, or one minute.

Example

```
WsGetfile("http://maps.googleapis.com/maps/api/staticmap?center=Costa  
Mesa,CA&zoom=14&size=400x400&sensor=false", "C:\Test\CMmap.png", 120000)
```

Parent topic: [Web Services actions](#)

WsGetValues

The WsGetValues action returns one or more values from a web service.

Syntax

```
bool WsGetValues (string url, string nsprefix, string nsuri, string source, string  
target, string format, string savefile)
```

Parameters

url

Required: the URL of the web service.

nsprefix

Optional: the prefix of the Namespace if you are using tags to obtain results.

nsuri

Optional: the Namespace URI. If a namespace has already been set, you do not need to set the URI.

source

Optional: the xpath or tag to obtain the value. The xpath/tag and target are set separately.

target

Optional: the target DCO. You can set the xpath/tag and target separately, and if no path is set then the entire result set is saved to the target.

format

Optional: the return format type. The default is XML.

savefile

Optional: save the response results to this file for reference.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Details

The value for the URL should include the end point. This is the only required value if the parameters and target values are set before calling the action.

Smart parameters are supported.

Example

Example 1 uses xpath with a namespace to obtain information from the web service:

```
WsGetvalues("http://api.worldbank.org/countries?format=xml", "wb",  
"http://www.worldbank.org", "//wb:countries/wb:country[5]/wb:name", "@B.Country",  
"", "C:\myfolder\myfile.xml")
```

Example 2 uses tag value to obtain the information:

```
WsGetvalues("http://www.webservices.net/CurrencyConvertor.asmx/ConversionRate?  
FromCurrency=USD&ToCurrency=GBP", "", "", "double", "@B.ConversionRate", "", "")
```

Example 3 shows the syntax for JSON:

```
WsSetParameter("symbols", "USD", true)  
    WsSetParameter("symbols", "GBP", true)  
    WsGetvalues("http://api.fixer.io/latest?", "", "", "", "", "JSON",  
"@PILOT(BATCHDIR)+/rates+.json")
```

Parent topic: [Web Services actions](#)

WsSetCredentials

Associates credentials that you send with web service calls for authentication.

Syntax

```
bool WsSetCredentials (string username, string password)
```

Parameters

username

The credential user name. Smart parameters are supported.

password

The credential password. Smart parameters are supported.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Details

Use this action with the [WsUploadData](#) or [WsUploadFile](#) actions.

Example

```
WsSetCredentials ("abcdef", "xyz123")
```

Parent topic: [Web Services actions](#)

WsSetHeader

Specify any header information about the response content.

Syntax

```
bool WsSetHeader (string name, string value, bool add)
```

Parameters

name

The name of the header. Smart parameters are supported.

value

The value for the specified header. Smart parameters are supported.

add

If the specified header already exists, then set this parameter to TRUE for the new value to be added to the existing value. If you set this parameter to FALSE, then the existing value will be replaced by the new value.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Example

```
WsSetHeader("Content-Type", "text/xml", true)
```

Parent topic: [Web Services actions](#)

WsSetNamespace

Specify the namespace associated with the web service response.

Syntax

```
bool WsSetNamespace (string prefix, string uri)
```

Parameters

prefix

The prefix for the namespace. Smart parameters are supported.

uri

The URI of the namespace. Smart parameters are supported.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Example

```
WsSetNamespace ("wb", "http://www.worldbank.org")  
WsGetvalues ("http://api.worldbank.org/countries?format=xml", "wb", "",  
"/wb:countries/wb:country[5]/wb:name", "@B.Country", "")
```

Parent topic: [Web Services actions](#)

WsSetParameter

Set the parameters required by the web service endpoint.

Syntax

```
bool WsSetParameter (string name, string value, bool add)
```

Parameters

name

The name of the parameter. Smart parameters are supported.

value

The value for the parameter. Smart parameters are supported.

add

If the specified parameter already exists, then set this parameter to **TRUE** for the new value to be added to the existing value. If you set the parameter to **FALSE**, then the existing value will be replaced by the new value.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Example

```
WsSetParameter("FromCurrency", "USD", false)
WsSetParameter("ToCurrency", "GBP", false)

WsGetvalues("http://www.websvc.net/CurrencyConvertor.asmx/ConversionRate", "",
"", "double", "@B.GBPConversionRate", "")
    WsSetParameter("ToCurrency", "EUR", false)

WsGetvalues("http://www.websvc.net/CurrencyConvertor.asmx/ConversionRate", "",
"", "double", "@B.EURConversionRate", "")
```

Parent topic: [Web Services actions](#)

WsSetResultItem

This action will associate the keys(xpath or tags) to the values from the response to the DCO objects to which you want to assign them. If the source already exists its corresponding target will be replaced with the new target value.

Syntax

```
bool WsSetResultItem (string source, string target)
```

Parameters

source

The xpath or tags to find the value in the result. Smart parameters are supported.

target

Destination item to assign the value.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Details

This action must be set before calling the response. Use this action if you want to get more than a single value from the response.

Example 1

```
WsSetParameter("FromCurrency", "USD", false)
    WsSetParameter("ToCurrency", "GBP", false)
    WsSetResultItem("double", "@B.ConversionRate")

WsGetvalues("http://www.webservices.net/CurrencyConvertor.asmx/ConversionRate?", "",
"", "", "", "")
```

Example 2

```
WsSetResultItem("@STRING(rates.USD)", "@B.EUR2USD")
    WsSetResultItem("@STRING(rates.GBP)", "@B.EUR2GBP")
    WsGetvalues("http://api.fixer.io/latest?", "", "", "", "", "JSON", "")
```

Parent topic: [Web Services actions](#)

WsSetTimeout

The timeout value specified with this action will be used for all subsequent requests to web services.

Syntax

```
bool WsSetTimeout (int timeout)
```

Parameters

timeout
Timeout value in milliseconds.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Details

If no timeout value is specified, the default value of 60000 (60 seconds) is used.

Example

```
WsSetTimeout(120000)
```

Parent topic: [Web Services actions](#)

WsUploadData

Upload data to a Webservice endpoint using a POST request.

Syntax

```
bool WsUploadData (string url, string uploadData, string sessionkeys, string format, string responsefile)
```

Parameters

url

Required: the URL of the web service.

uploadData

Required: the data to upload.

sessionkeys

Optional: comma-separated list of session key names.

format

Optional: return format type. The default is xml.

responsefile

Optional: save the response results to this file for later use.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Details

Smart parameters are supported for all parameters.

Example

The example shows how to use wTM endpoints to log on and create a batch:

```
WsSetHeader("@STRING(Content-Type)", "@STRING(application/xml)", false)
WsUploadData("http://localhost:8010/ServicewTM.svc/Session/Logon",
  "Demo901adminladmin", "", "xml", "C:\myfolder\logon.xml")
WsUploadData("http://localhost:8010/ServicewTM.svc/Queue/CreateBatch",
  "Demo901Demo_SingleTIFFs", "Set-Cookie", "", "C:\myfolder\createbatch.xml")
```

Parent topic: [Web Services actions](#)

WsUploadFile

Uploads a file to a web service.

Syntax

```
bool WsUploadFile (string url, string filename, string method, int timeout)
```

Parameters

url

Required: the URI of the resource to receive the file. Smart parameters are supported.

filename

Required: the file to send to the resource. Smart parameters are supported.

method

Optional: the file upload method. By default, 'POST' is used for http and 'STOR' for ftp.

timeout

Optional: the timeout value in milliseconds before abandoning the request. Smart parameters are supported.

Returns:

True, if the action succeeds. Otherwise, **False**.

Level

All levels.

Details

The timeout value specified for this operation will override any timeout value set elsewhere. The default timeout value is 60000 milliseconds, or one minute. If no method is specified, the default is "POST" for http and "STOR" for ftp.

Example

```
WsUploadFile("http://server/service", "C:\Test\myfile.jpg", "", "120000")
```

Parent topic: [Web Services actions](#)

Zones actions

Use the Zones actions to work with the zones that define the position of each field on the page.

You can read zone information from a fingerprint (CCO) file, update the zone position information in the runtime hierarchy, and locate the recognition text for a specific zone. You can also assign values to fields in the runtime hierarchy, locate repeating data blocks, and more.

- [AdjustZonesToImageOffset](#)
Offsets fields on the current image in response to zone criteria in the CCO file of the source page - after the image has been offset.
- [AnchorPage](#)
Finds the Anchor field on a source page, and uses the Anchor field's coordinates to locate and offset the page's other zoned fields.
- [CalculateLocalOffset](#)
Calculates the X and/or Y offset amount for the calling field.
- [CreateBlockCCO](#)
Creates a temporary in memory CCO object, containing only the words and lines in the calling fields zone position - using the source page's CCO file.
- [FindBlocks_WhiteSpace](#)
Uses a vertical white space (pixels) to find blocks of data within the current source page. Returns each block's position assigned to a series of repeating fields based on the first child field of the calling object.

- [FindDataBlocks](#)
Uses Start and End key words to find blocks of data within the current source page.
- [FindRegExBlocks](#)
Uses a Regular Expression to find blocks of data within the current source page. Returns the position of each block that is assigned to a series of repeating fields based on the first child field of the calling object.
- [FindZoneLineItems](#)
Uses settings for zones to assemble a portion of the current page, limited to Line Item Detail.
- [GetZoneText](#)
Retrieves the text in a zoned object of the current page.
- [InheritParentPosition](#)
Provides the bound child object of the Document Hierarchy with the zone parameters of a parent object identified by the parameter.
- [LoadBlockCCO](#)
Loads the CCOBlock set up by a previous CreateBlockCCO action. That action assigns the block's location to the CCOBlock variable of the current Page object.
- [LoadZones](#)
Same as the ReadZones action, except that it loads position information for the specified fingerprint ID.
- [MCCOPositionAdjust](#)
Combines extra pages of a multi-page document into the CCO file for the first page.
- [MergeZones](#)
Merges the zone from calling field with zone of DCO fields passed as smart parameters.
- [PadZone](#)
Pads the zone by the value passed. Number of CSV passed values varies padding value by vector.
- [PopulateZNField](#)
Puts the recognition data from the CCO file that lies in the zone boundaries of the field into the current field in the page data file.
- [PopulateZNLineItemField](#)
Populates the page data file with the recognized value in the zone for the current line item child field. Assign this action to each line item child field in the document hierarchy.
- [ReadZones](#)
Loads the position information for the current object and its children from the document hierarchy (setup DCO). Adjusts the position of each object by using any Image_Offset value.
- [RegisterPage](#)
Locates specially marked fields and adjusts their vertical zone positions to compensate for any drift.
- [ScanDetails](#)
Searches a line item grid object for line items. You assign this action to the grid region in the document hierarchy.
- [ScanDetailsByLines](#)
Searches a line item grid object for line items, where each line item consists of the specified number of rows.
- [ScanDetailsByVSpace](#)
Searches a line item grid object for line items, where each line item is defined as the specified height in pixels.
- [ScanLineItem](#)
Searches a line item object for fields. You assign this action to each line item in the document hierarchy.
- [SetEOL](#)
Sets the End of Line character that will be used to separate data from a zone that contains multiple lines of text.
- [SetEOL_CRLF](#)
Sets the End Of Line character that are used to separate data from a zone with multiple lines of text.
- [ZoneBOTTOM_ImageBottom](#)
Uses the lower boundary of the current image to specify the bottom boundary of the zone position of the

- current field in the page data file.
- [ZoneBOTTOM_LowerBound](#)
Uses the lower boundary of the current word that is located by using the actions in the Locate library to specify the bottom boundary of the zone position of the current field in the page data file.
 - [ZoneBOTTOM_UpperBound](#)
Uses the upper boundary of the current word that is located by using the actions in the Locate library to specify the bottom boundary of the zone position of the current field in the page data file.
 - [ZoneImage_SaveAs](#)
Saves the current Objects Zone area of an image as a separate Image file.
 - [ZoneLEFT_ImageLeft](#)
Uses the left boundary of the current image to specify the left boundary of the zone position of the current field in the page data file.
 - [ZoneLEFT_LeftBound](#)
Uses the left boundary of the current image to specify the left boundary of the zone position of the current field in the page data file.
 - [ZoneLEFT_RightBound](#)
Uses the right boundary of the current image to specify the left boundary of the zone position of the current field in the page data file.
 - [ZoneRIGHT_ImageRight](#)
Uses the right boundary of the current image to specify the right boundary of the zone position of the current field in the page data file.
 - [ZoneRIGHT_LeftBound](#)
Uses the left boundary of the current word* to specify the right boundary of the zone position of the current field in the page data file.
 - [ZoneRIGHT_RightBound](#)
Uses the right boundary of the current word* to specify the right boundary of the zone position of the current field in the page data file.
 - [ZoneTOP_ImageTop](#)
Uses the top boundary of the current image to specify the top boundary of the zone position of the current field in the page data file.
 - [ZoneTOP_LowerBound](#)
Uses the lower boundary of the current word that is located by using the actions in the Locate library to specify the top boundary of zone position of the current field in the page data file.
 - [ZoneTOP_UpperBound](#)
Uses the upper boundary of the current word that is located by using the actions in the Locate library to specify the top boundary of the zone position of the current field in the page data file.

Parent topic: [Global actions](#)

AdjustZonesToImageOffset

Offsets fields on the current image in response to zone criteria in the CCO file of the source page - after the image has been offset.

Syntax

```
bool AdjustZonesToImageOffset ()
```

Parameters

None.

Returns

False if the action cannot locate the CCO file of the current page or an image offset value does not exist for the current page. Otherwise, True.

Level

Page level.

Details

Offsets fields on the current image in response to zone criteria in the CCO file of the source page - after the image has been offset.

Example

```
AdjustZonesToImageOffset ()
```

Parent topic: [Zones actions](#)

AnchorPage

Finds the Anchor field on a source page, and uses the Anchor field's coordinates to locate and offset the page's other zoned fields.

Syntax

```
bool AnchorPage ()
```

Parameters

None.

Returns

False if the action cannot find the Anchor field on the page. Otherwise, True.

Level

Page level.

Details

Finds the Anchor field on a source page, and uses the coordinates of the Anchor field to locate and offset the other zoned fields on the page.

Example

```
AnchorPage ()
```

Parent topic: [Zones actions](#)

CalculateLocalOffset

Calculates the X and/or Y offset amount for the calling field.

Syntax

```
bool CalculateLocalOffset (StrParam)
```

Parameters

X and/or Y

The action uses these parameters to calculate a new parent page Image_Offset value by comparing the fingerprint zone for the calling field against the current field zone.

Returns

Always True.

Level

Field only.

Details

Calculates the X and/or Y offset amount for the calling field.

Example

```
CalculateLocalOffset ("XY")
```

Parent topic: [Zones actions](#)

CreateBlockCCO

Creates a temporary in memory CCO object, containing only the words and lines in the calling fields zone position - using the source page's CCO file.

Syntax

```
bool CreateBlockCCO ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Useful when searching for images where the data is located asymmetrically across the page. Using this action focuses the CCO to only look at the words and lines within the defining zone in all future searches for this page.

This does not affect the run time CCO. Reloading the page reloads the CCO bound to the page, releasing this temporary CCO.

Example

```
CreateBlockCCO()
```

Parent topic: [Zones actions](#)

FindBlocks_WhiteSpace

Uses a vertical white space (pixels) to find blocks of data within the current source page. Returns each block's position assigned to a series of repeating fields based on the first child field of the calling object.

Syntax

```
bool FindBlocks_WhiteSpace (Strparam)
```

Parameters

A single parameter indicating the number of pixel between lines.

Returns

False if the action cannot divide or create any child field/zones. Otherwise, True.

Level

Field having 1 child field.

Details

Creates a child field with the position of each zone divided out of the CCO of the calling page.

Example

```
FindBlocks_WhiteSpace ("27")
```

Parent topic: [Zones actions](#)

FindDataBlocks

Uses Start and End key words to find blocks of data within the current source page.

Syntax

```
bool FindDataBlocks (Strparam)
```

Parameters

Key word Start Value, and its End Value.

Returns

True if the action can locate the Data block indicated by the parameter. Otherwise, False.

Level

Page level.

Details

Uses Start and End key words to find blocks of data within the current source page. Returns each block's position assigned to a series of repeating fields based on the first child field of the calling object. If the Fingerprint file (.cco) of the page has a Line position be sure to use action GoFirstLine() to set the Line position to the first word on the first line in the current zone.

Example

```
GoFirstLine ()  
FindDataBlocks ("FROM, THRU")
```

Parent topic: [Zones actions](#)

FindRegExBlocks

Uses a Regular Expression to find blocks of data within the current source page. Returns the position of each block that is assigned to a series of repeating fields based on the first child field of the calling object.

Syntax

```
bool FindRegExBlocks (Strparam)
```

Parameters

Regular Expression that contains the data block's Start Value, and its End Value. Parameter is a two part comma separated value of the Values to look for in the current CCO.

1. StartValue
2. EndValue

Start and End values can be adjusted up or down by xLines using the 3rd (adj top) and 4th (adj bottom) CSV positions.

Returns

True if the action can locate the Data block indicated by the parameter. Otherwise, False.

Level

Page level.

Details

Uses a Regular Expression to find blocks of data within the current source page.

Attention: Locate.rra is required for this action.

Example

```
FindRegExBlocks ("/bFROM/b, /bTHRU/b")
```

Parent topic: [Zones actions](#)

FindZoneLineItems

Uses settings for zones to assemble a portion of the current page, limited to Line Item Detail.

Syntax

```
bool FindZoneLineItems (string createSubfields, string createTempCCO, string subfieldFillType, string offsetPercent, string intersectionRatio)
```

Parameters

1. True or False
 - o True if the action is to create Line-item fields and their sub-fields.
 - o False if the action is to create Line-item fields only.
2. True or False (OPTIONAL: defaults to True)
 - o True if the action is to analyze both the calling zone and the word and line relationships based only on the zone contents that are found in the search.
 - o False if the action is to use the existing word and line relationships (CCO for image).
3. 1 or 2 (OPTIONAL: Fill Sub fields option - defaults to 1)
 - o 1 - fill by CCO word; if the CCO words overlap by a certain intersection value then fill sub field with its value. Also has sub setting of a colon followed by a group words value. For example, 1:1.5 treats words 1.5 character spaces apart or closer as one word.
 - o 2 - fill by CCO character; if the CCO characters overlap by a certain intersection value then fill sub field with the character value.
4. 0 to 100 (OPTIONAL: Percent offset adjustment - defaults to 0) - moderates the flexibility for the left offset based on the overall length of the line. Use larger values for shorter line items.
5. .01 to .99 (OPTIONAL: Intersection Ratio - defaults to .25) - changes the intersection ratio of Words and Lines used to determine creation and value of Lineitems and their subfields.

Returns

False if there is no zone or position information for the Line Item block and its subfields, or if the parent block does not contain Line Item children. Otherwise, True.

Level

Field having 1 child field.

Details

Uses settings for a previously-established zone covering a block of Line Item Detail, and zones for individual Line Items, to assemble a portion of the current page, limited to Line Item Detail.

Example

```
FindLineItems ("True", "", "", "", "")
```

This action retrieves zonal information from the fingerprint about a block of Line Item Detail, individual rows, and their fields and subfields. The action applies this data to the full-length current page. It creates a file with an extension of cco (fingerprint format file) containing just Line Item Detail for use during batch processing. This file can be discarded after batch processing has finished.

Parent topic: [Zones actions](#)

GetZoneText

Retrieves the text in a zoned object of the current page.

Syntax

```
bool GetZoneText ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Retrieves the text in a zoned object of the current page.

Example

```
GetZoneText ()
```

This action assumes that you have established zone parameters for this Field object of the Document Hierarchy.

Parent topic: [Zones actions](#)

InheritParentPosition

Provides the bound child object of the Document Hierarchy with the zone parameters of a parent object identified by the parameter.

Syntax

```
bool InheritParentPosition (StrParam)
```

Parameters

Case sensitive string value of the name of the parent object. Details, for example, is the name of a parent Field object in the Invoice application. If the bound object is LINEITEM, the action provides the zone parameters of

the parent to this child object.

Returns

False if the action cannot locate the parent field. Otherwise, True.

Level

Field level.

Details

Provides the bound child object (a subfield of another field) of the Document Hierarchy with the zone parameters of a parent object identified by the parameter.

Example

```
InheritParentPosition("Details")
```

Parent topic: [Zones actions](#)

LoadBlockCCO

Loads the CCOBlock set up by a previous CreateBlockCCO action. That action assigns the block's location to the CCOBlock variable of the current Page object.

Syntax

```
bool LoadBlockCCO ()
```

Parameters

None.

Returns

False if the block cannot be found. Otherwise, True.

Level

Page level.

Details

Loads the CCOBlock set up by a previous CreateBlockCCO action. That action assigns the block's location to the current Page object's CCOBlock variable.

Example

```
CreateBlockCCO ()  
LoadBlock ()
```

Parent topic: [Zones actions](#)

LoadZones

Same as the ReadZones action, except that it loads position information for the specified fingerprint ID.

Syntax

```
bool LoadZones (StrParam)
```

Parameters

Fingerprint ID.

Returns

Always True.

Level

Page or Field level.

Details

Loads position information for each node in the calling object and children. Pre-adjusts these values based on offset information stored in an *Image_Offset* variable at any node level.

Position information is based on the setup DCO position for the fingerprint ID passed as a parameter. The offset value is applied to all child objects of a node where an *Image_Offset* variable is found; unless overwritten by a child node also having an *Image_Offset* value to apply.

Parent topic: [Zones actions](#)

MCCOPositionAdjust

Combines extra pages of a multi-page document into the CCO file for the first page.

Syntax

```
bool MCCOPositionAdjust ()
```

Parameters

None.

Returns

False if the current document does not consist of more than one source page, or if a page to be merged did not have an associated CCO file created. Otherwise, True.

Level

Page level.

Details

MCCOPositionAdjust is an important action for applications using merged multiple page documents. It normalizes position coordinates populated to the DCO when the data is generated from a document using a merged CCO. The coordinates are adjusted relative to the page and also sets the DCO ImageFile property to the name of the TIFF that the data was found on.

Attention: All files to be merged must have an associated CCO file.

This action is used in conjunction with action MergeCCO_ByType from the Autodoc library in a prior ruleset.

Example

```
MCCOPositionAdjust()
```

Parent topic: [Zones actions](#)

MergeZones

Merges the zone from calling field with zone of DCO fields passed as smart parameters.

Syntax

```
bool MergeZones (string SmartParameters)
```

Parameters

A CSV of smart parameter strings indicating the DCO fields to merge zones with.

Returns

False if the calling field does not have a valid position value. Otherwise, True.

Level

Field level.

Details

Merges calling field's position with each passed dco position.

Example

```
MergeZones("@P\5PAddTel,@P\5PAddZip,@P\2PatName")
```

Parent topic: [Zones actions](#)

PadZone

Pads the zone by the value passed. Number of CSV passed values varies padding value by vector.

Syntax

```
bool PadZone (string SmartParameters)
```

Parameters

A CSV of smart parameter strings that indicate the pixels value by which to pad the calling field.

Returns

False, if the calling field does not have a valid position that is not a zero.

False, if there are zero or more than four CSV parameters.

False, if the Smart Parameter value returns as a non-numeric.

Otherwise, True.

Level

Field level.

Details

Pads calling field's position with each value passed. Number of passed values varies padding vector as follows:

1. Pads Left, Top, Right, and Bottom by this value (that is, expand if positive)
2. Pads as X,Y (X: pad to Left and Right, Y: pad to Top and Bottom)
3. Pads as Left, Top, Right, and Bottom. (missing values are treated as zero).
4. Pads as Left, Top, Right, and Bottom.

Example

```
PadZone("15") Pads 15 pixels to the Left, Right, Top and Bottom position.  
PadZone("5,10") Pads 5 pixels to the Left and Right position, and 10 to the Top  
and Bottom position.  
PadZone("-5,0,12,25") Pads -5 pixels to the Left, 0 to the Right, 12 to the Top  
and 25 to the Bottom position.
```

Parent topic: [Zones actions](#)

PopulateZNField

Puts the recognition data from the CCO file that lies in the zone boundaries of the field into the current field in the page data file.

Syntax

```
bool PopulateZNField ()
```

Parameters

None.

Returns

True if a value is found. Otherwise, False.

Level

Field level.

Details

This action populates the field value with the full-page recognition results (in the CCO) that fall within the boundaries of the field zone.

This action should be used with fields such as the Total or Number field in the Invoice application.

Example

```
PopulateZNField()
```

Parent topic: [Zones actions](#)

PopulateZNLineItemField

Populates the page data file with the recognized value in the zone for the current line item child field. Assign this action to each line item child field in the document hierarchy.

Syntax

```
bool PopulateZNLineItemField ()
```

Parameters

None.

Returns

True if a value is found or if the calling field has no position information. Otherwise, False.

Level

Field level.

Details

Populates the fingerprint's Data file with the recognized value contained inside the zone of a child Field object of a LINEITEM parent field. This action should only be used with subfields of the LINEITEM field (ItemID, ItemDesc, Quantity, Price) in the Invoices application.

Example

```
PopulateZNLineItemField()
```

Parent topic: [Zones actions](#)

ReadZones

Loads the position information for the current object and its children from the document hierarchy (setup DCO). Adjusts the position of each object by using any *Image_Offset* value.

Syntax

```
bool ReadZones ()
```

Parameters

None.

Returns

Always True.

Level

Page or Field level.

Details

Loads position information for each node in the calling object and its children. Pre-adjusts these values based on offset information stored in an *Image_Offset* variable at any node level.

Position information is based on the setup DCO position for the fingerprint ID of the parent page. The offset value is applied to all child objects of a node where an *Image_Offset* variable is found; unless overwritten by a child node also having an *Image_Offset* value to apply.

Parent topic: [Zones actions](#)

RegisterPage

Locates specially marked fields and adjusts their vertical zone positions to compensate for any drift.

Syntax

```
bool RegisterPage ()
```

Parameters

None.

Returns

Always True.

Level

Page level.

Details

RegisterPage adjusts the y positions of special fields to coordinate zone positions with the current word location of the fingerprint. Searches the calling page object for run time field names beginning with the word 'Anchor' (case sensitive). Of these fields a search is performed for variables named Word followed by the calling fingerprint ID (example: 'Word555'). The value of the variable retrieved is search for in the current pages CCO, and if found the field's zone position is 'y' axis adjusted for any word 'drift' for the given field value.

Example

```
RegisterPage ()
```

Parent topic: [Zones actions](#)

ScanDetails

Searches a line item grid object for line items. You assign this action to the grid region in the document hierarchy.

Syntax

```
bool ScanDetails ()
```

Parameters

None.

Returns

True if the bound Field object contains lines of data. Otherwise, False.

Level

Field level.

Details

Searches the DETAILS Field object's zone for instances of a LINEITEM Field object. Captures the data in the rows of a Line Item table, row by row.

This action captures all potential LINEITEM rows within the parent DETAILS field, even rows that may not fit your criteria for content.

Attention: You must run additional rulesets to delete ineligible Line Items. In the APT application, these are the Clean and Filter rulesets.

Example

```
ScanDetails ()
```

Parent topic: [Zones actions](#)

ScanDetailsByLines

Searches a line item grid object for line items, where each line item consists of the specified number of rows.

Syntax

```
bool ScanDetailsByLines (sParam)
```

Parameters

Number of lines in each lineitem.

Returns

True if the bound Field object contains lines of data. Otherwise, False.

Level

Field level.

Details

Searches the DETAILS Field object's zone for instances of a LINEITEM Field object. Captures the data based on the number of lines (parameter) in a Line Item table, row by row.

This action captures all potential LINEITEM rows (consisting of 2 lines each) in the parent DETAILS field, even rows that may not fit your criteria for content.

Attention: You must run additional rulesets to delete ineligible Line Items. In the Invoices application, these are the Clean and Filter rulesets.

Example

```
ScanDetailsByLines ("2")
```

Parent topic: [Zones actions](#)

ScanDetailsByVSpace

Searches a line item grid object for line items, where each line item is defined as the specified height in pixels.

Syntax

```
bool ScanDetailsByVSpace (sParam)
```

Parameters

Number of vertical pixels in each lineitem.

Returns

True if the bound Field object contains lines of data. Otherwise, False.

Level

Field level.

Details

Searches the DETAILS Field object's zone for instances of a LINEITEM Field object. Captures the data based on the number of vertical pixels (parameter) in a Line Item table, row by row.

Example

```
ScanDetailsByLines ("45")
```

This action capture all potential LINEITEM rows (consisting of 45 pixels lines each) in the parent DETAILS field, even rows that might not fit your criteria for content.

Attention: You must run additional rulesets to delete ineligible Line Items. In the Invoices application, these are the Clean and Filter rulesets.

Parent topic: [Zones actions](#)

ScanLineItem

Searches a line item object for fields. You assign this action to each line item in the document hierarchy.

Syntax

```
bool ScanLineItem ()
```

Parameters

None.

Returns

False if not called from a Field. Otherwise, True.

Level

Field level.

Details

This action captures each subfield within the parent LINEITEM row.

Example

```
ScanLineItem ()
```

Parent topic: [Zones actions](#)

SetEOL

Sets the End of Line character that will be used to separate data from a zone that contains multiple lines of text.

Syntax

```
bool SetEOL (bParam)
```

Parameters

The End of Line separator character.

Returns

Always True.

Level

All.

Details

If this action is not used, the default character is a space.

Example

```
SetEOL ("|")
```

This example sets the End of Line character to the '|' (pipe) character. A capture zone with two lines of text contains the captured value separated by this new character.

Parent topic: [Zones actions](#)

SetEOL_CRLF

Sets the End Of Line character that are used to separate data from a zone with multiple lines of text.

Syntax

```
bool SetEOL_CRLF ()
```

Parameters

None.

Returns

Always True.

Level

All.

Details

Sets the End Of Line character that is used to separate data from a zone with multiple lines of text to the carriage return and Line Feed characters: ASCII values 13 and 10.

Example

```
SetEOL_Crlf()
```

This example sets the End of Line character to the Carriage Return and Line Feed characters. A capture zone with two lines of text contains the captured value separated by these characters.

Parent topic: [Zones actions](#)

ZoneBOTTOM_ImageBottom

Uses the lower boundary of the current image to specify the bottom boundary of the zone position of the current field in the page data file.

Syntax

```
bool ZoneBOTTOM_ImageBottom ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Uses the bottom boundary of the current field's IMAGE to define the bottom edge of the field's ZONE. Attention: An image of a field is often larger than a zone of a field, and almost always larger than the field's word.

Example

```
ZoneBOTTOM_ImageBottom()
```

Parent topic: [Zones actions](#)

ZoneBOTTOM_LowerBound

Uses the lower boundary of the current word that is located by using the actions in the Locate library to specify the bottom boundary of the zone position of the current field in the page data file.

Syntax

```
bool ZoneBOTTOM_LowerBound ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Uses the lower boundary of the current field's ZONE to specify the bottom of the field's WORD. This can result in a taller or shorter word, depending on the location of the zone's boundary.

Example

```
ZoneBOTTOM_LowerBound()
```

If you find that an unacceptably high percentage of values for a particular field are entered a line too high, consider the use of this action in a follow-up Locate rule to search for the misplaced value.

Parent topic: [Zones actions](#)

ZoneBOTTOM_UpperBound

Uses the upper boundary of the current word that is located by using the actions in the Locate library to specify the bottom boundary of the zone position of the current field in the page data file.

Syntax

```
bool ZoneBOTTOM_UpperBound ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Uses the lower boundary of the current field's ZONE to specify the top of the field's WORD. This can result in a taller or shorter word, depending on the location of the zone's lower boundary.

Example

```
ZoneBOTTOM_UpperBound()
```

This action is helpful when experience shows that a certain number of values in this field extend beyond the word's upper limit. A follow-up Locate rule with this action can find those values.

Parent topic: [Zones actions](#)

ZoneImage_SaveAs

Saves the current Objects Zone area of an image as a separate Image file.

Syntax

```
bool ZoneImage_SaveAs (StrFileName)
```

Parameters

A string that defines the file name.

You can use these options to format the file name:

- **+@BATCHID** : Adds the BatchID to the Zone Image File Name
- **+@ID** : Adds the Object ID to the Zone Image File Name
- **+@STATUS** : Adds the Object Status to the Zone Image File Name
- **+@TYPE** : Adds the Object Type to Zone Image File Name
- **+@DATE+mm/dd/yyyy** : Adds a Date Stamp to the Zone Image File Name, the required trailing date format argument shows as the default, also '+' can be used
- **+@TIME+HH:MM:SS** : Adds a Time Stamp Value to the Zone Image File Name, the required trailing time format argument shows as the default, also '+' can be used
- **+@VALUE** : Adds Object Text to Zone Image File Name
- **+#name** : Appends the value of a child name to the Image File Name

Attention: **+#name** appends the value of child name to the image name.

Returns

False if an image cannot be saved. Otherwise, True.

Level

Page or Field level.

Details

Saves the current Objects Zone area of an image as a separate Image file. The Zone Image file is always placed in the Batches directory; saving to the current page creates a multi-page TIFF file with the additional image as its zone.

Example

```
ZoneImage_SaveAs ("SAMMY+@TYPE+@DATE+JJJ")
```

As part of a rule bound to the Details Field object of the Document Hierarchy, the action produces this Image File Name: ...\\SAMMYDETAILS243.tif.

Parent topic: [Zones actions](#)

ZoneLEFT_ImageLeft

Uses the left boundary of the current image to specify the left boundary of the zone position of the current field in the page data file.

Syntax

```
bool ZoneLEFT_ImageLeft ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Uses the left boundary of the current field's image to define the left edge of the field's zone. This can extend or contract the width of the zone, depending on the placement of the left side of the image.

Attention: A field's image is often larger than a field's zone, and almost always larger than the field's word.

Example

```
ZoneLEFT_ImageLeft ()
```

Parent topic: [Zones actions](#)

ZoneLEFT_LeftBound

Uses the left boundary of the current image to specify the left boundary of the zone position of the current field in the page data file.

Syntax

```
bool ZoneLEFT_LeftBound ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Uses the left boundary of the current field's word to define the left edge of the field's zone. This extends the width of the zone to accommodate the current word.

Example

```
ZoneLEFT_LeftBound()
```

If the field's recognized values reach beyond the zone's limits, you can use this action to expand the zone appropriately.

Parent topic: [Zones actions](#)

ZoneLEFT_RightBound

Uses the right boundary of the current image to specify the left boundary of the zone position of the current field in the page data file.

Syntax

```
bool ZoneLEFT_RightBound ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Uses the right boundary of the current field's word to designate the left edge of the field's zone. This extends the width of the zone to accommodate the current word.

Example

```
ZoneLEFT_RightBound()
```

Attention: The image of a field is often larger than the field's zone, and almost always larger than the field's word.

Parent topic: [Zones actions](#)

ZoneRIGHT_ImageRight

Uses the right boundary of the current image to specify the right boundary of the zone position of the current field in the page data file.

Syntax

```
bool ZoneRIGHT_ImageRight ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Uses the right boundary of the current field's image to define the right edge of the field's zone. This can extend or contract the width of the zone, depending on the placement of the right side of image.

Attention: A field's image is often larger than a field's zone, and almost always larger than the field's word.

Example

```
ZoneRight_ImageRight()
```

Parent topic: [Zones actions](#)

ZoneRIGHT_LeftBound

Uses the left boundary of the current word* to specify the right boundary of the zone position of the current field in the page data file.

Syntax

```
bool ZoneRIGHT_LeftBound ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Uses the left boundary of the current field's word to designate the right edge of the field's zone. This action also excludes the word from the zone.

Example

ZoneRight_LeftBound()

This action can clean the zone by eliminating the word it contains.

Parent topic: [Zones actions](#)

ZoneRIGHT_RightBound

Uses the right boundary of the current word* to specify the right boundary of the zone position of the current field in the page data file.

Syntax

```
bool ZoneRIGHT_RightBound ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Uses the right boundary of the current field's word to designate the right edge of the field's zone.

This is a convenient action if a percentage of the field's entered values extend beyond the right edge of the zone.

Example

```
ZoneRIGHT_RightBound()
```

Parent topic: [Zones actions](#)

ZoneTOP_ImageTop

Uses the top boundary of the current image to specify the top boundary of the zone position of the current field in the page data file.

Syntax

```
bool ZoneTOP_ImageTop ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

Uses the upper boundary of the current field's image to designate the top edge of the field's zone. This can extend or contract the width of the zone, depending on the placement of the left side of the image.

Example

```
ZoneTOP_ImageTop ()
```

Parent topic: [Zones actions](#)

ZoneTOP_LowerBound

Uses the lower boundary of the current word that is located by using the actions in the Locate library to specify the top boundary of zone position of the current field in the page data file.

Syntax

```
bool ZoneTOP_LowerBound ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

This action uses the upper boundary of the current field's zone to indicate the bottom of the field's word. This action can expand or contract the height of the zone, depending on the placement of the word's lower boundary within the current page.

Example

```
ZoneTOP_LowerBound ()
```

If you find that a high percentage of values for a particular field are entered a line too high, you can use this action in a follow-up Locate rule to search for the misplaced value.

Parent topic: [Zones actions](#)

ZoneTOP_UpperBound

Uses the upper boundary of the current word that is located by using the actions in the Locate library to specify the top boundary of the zone position of the current field in the page data file.

Syntax

```
bool ZoneTOP_UpperBound ()
```

Parameters

None.

Returns

Always True.

Level

Field level.

Details

This action is helpful when you know that a certain number of values in this field will be shorter than the zone's height. A follow-up Locate rule with this action will find those values.

Example

```
ZoneTOP_UpperBound ()
```

Parent topic: [Zones actions](#)

Application specific actions

The Datacap applications use actions that are specific to them.

- [Medical Claims actions](#)
The following are actions specific to the Medical Claims application.
- [Datacap Accounts Payable actions](#)
Use the Datacap Accounts Payable actions when you work with batches of invoices on APT.

Parent topic: [Action library summaries](#)

Medical Claims actions

The following are actions specific to the Medical Claims application.

- [4010Common](#)
- [4010Institutional](#)
- [4010Professional](#)
- [5010Common](#)
- [5010Institutional](#)

- [5010Professional](#)
- [MC_Identify](#)
Use the MC_Identify actions to identify claim forms in a batch.
- [MC_Validation](#)

Parent topic: [Application specific actions](#)

4010Common

The set of *4010Common* actions are common to both Institutional and Professional medical claim forms.

Action	Description
Load4010 Settings	Load settings from the .ini file that is passed as a parameter. You can change default values and customize how the 4010 form is built. Smart parameter enabled.
Merge4010	Zips the 4010 file path, file name, and file extension settings into one file. Smart parameter enabled.

Parent topic: [Medical Claims actions](#)

4010Institutional

This action is specific to the Institutional 4010 form.

Action	Description
Institutional_4010	Used to build a single 4010 file for each Institutional claim. Use the action on a page or a document. The action allows you to create a single export page file for each document.

Parent topic: [Medical Claims actions](#)

4010Professional

This action is specific to the Professional 4010 form.

Action	Description
Professional_4010	Used to build a single 4010 file for each Professional claim. Use the action on a page or a document. The action allows you to create a single export page file for each document.

Parent topic: [Medical Claims actions](#)

5010Common

The set of *5010Common* actions are common to both Institutional and Professional medical claim forms.

Action	Description
Load5010 Settings	Load settings from the .ini file that is passed as a parameter. You can change default values and customize how the 5010 form is built. Smart parameter enabled.
Merge5010	Zips the 5010 file path, file name, and file extension settings into one file. Smart parameter enabled.

Parent topic: [Medical Claims actions](#)

5010Institutional

This action is specific to the Institutional 5010 form.

Action	Description
Institutional_5010	Used to build a single 5010 file for each Institutional claim. Use the action on a page or a document. The action allows you to create a single export page file for each document.

Parent topic: [Medical Claims actions](#)

5010Professional

This action is specific to the Professional 5010 form.

Action	Description
Professional_5010	Used to build a single 5010 file for each Professional claim. Use the action on a page or a document. The action allows you to create a single export page file for each document.

Parent topic: [Medical Claims actions](#)

MC_Validation

The actions in the *MC_Validation* library are used specifically with Professional and Institutional claim forms. The actions allow for the manipulation of batches, batch hierarchies, and data.

Action	Description
AddCenturyTo2DigitYear	Converts two-digit Year values to four-digit Year values.
AddToDetailErrorMessage	Adds the value to the existing value for the page variable ErrorMessage.
AddToErrorMessage	Adds the value to the existing value for the page variable ErrorMessage.
CalculateHCFALineCharges	Calculates charges for HCFA service lines.
CalculateResult	Determines if the calculation provided by the parameter is True or False.
CalculateUB	UB04 action that determines if the amounts in the 47ttchg fields sum correctly.
CalculateUBLineCharges	Calculates charges for UB service lines.
CheckDocID	Checks document IDs and updates them to the proper format.
ClearErrorMessage	Clears the value of the page variable ErrorMessage.
CommonParseAddresses	Parses addresses in certain fields in HCFA and UB04 forms into appropriate subfields.
CommonValAddresses	Validates address values.
ConvertHyphen	Removes spaces, commas and hyphens from the current field.
FilterPID	Filters qualifier from attending physician field for UB04 claims.
FormatFieldLengths	Truncates length and sets the last character to a low confidence recognition of the field.
InheritSnippets	Assigns the snippet position information of the current Field object to the Field objects specified in the parameter.

Action	Description
MC_ReadZones	Adjusts autofield based OMR field zone positions on the calling page.
Parse31aPhSig	Parses field 31aPhSig of the HCFA application.
Parse58ainsnm	Parses field 58ainsnm of the UB04 application.
Parse58binsnm	Parses field 58binsnm of the UB04 application.
Parse58cinsnm	Parses field 58cinsnm of the UB04 application.
Parse82name	Parses field 82name of the UB04 application.
Parse83aname	Parses field 83aname of the UB04 application.
Parse83bname	Parses field 83bname of the UB04 application.
ParseLastFirstIniNames	Parses the name information in the first line of an address superfield.
ParseNDC	An action that detects and parses NDC data elements from the calling field value.
ParseUB_Eighties	Special action to parse fields 82 (Attending Physician ID) and 83 (Other Physician ID) of the UB92 application.
PopulateFromField	Copies the value from the field specified by the parameter into the current field.
SetConf	Sets confidence string for a field.
SetOriginalTIF	Replaces current working TIF file.
StripTrailingAlpha	Removes all alpha characters from the captured value, except any in the first character position.
TransformLI	Remaps the Line Item Table fields into a hierarchical structure.
UpdateCredentialList	
ValidateNPI	Validates the NPI value by evaluating the 10 digits in the value uses a modified LUHN checkdigit algorithm.
ValidateStateMil	Checks to see if the value in the field represented by the bound Field object is a valid two-character State abbreviation.
ValProcedureCode	Validates the Procedure Code fields of a HCFA-1500 form.
ValRequiredGroup	Checks that all fields in a designated group are filled with data.

Parent topic: [Medical Claims actions](#)

Datacap Accounts Payable actions

Use the Datacap Accounts Payable actions when you work with batches of invoices on APT.

Datacap Accounts Payable actions are divided into the following categories:

Category	Description
Localization	Detects the localization and decimal separator settings of a workstation and changes the rules execution and data for the documents that you are processing if needed.
Custom	Customizes the fields in the invoice image to accommodate locale requirements and other things.

Category	Description
Concatenate line values	Merges line item values into page variable and page field values into document variables.
Documents	Creates and manages the documents in the batch.
FlexID	Starts the FlexID panel to manually identify the pages in the document batch.
Intellocate learning	Determines the new zones in the invoice image and adds these zones to the DCO.
Page ID	Separates and processes the page objects in the documents.
PreVerify setup	Sets the label variable that is used by Datacap Web Client and Datacap Desktop to display the label on each field.
Redaction	Redacts selected information in your invoice image for security purposes.

Embedded help is provided in Datacap Studio for all of the Datacap Accounts Payable actions. To access the embedded help, select an action in the Actions Library tab and click information. For detailed information, including information about parameters, refer to the embedded help.

- [APT_Localization](#)
Use the APT_Localization actions to work with currency values when you process documents generated from a locale that is different from the locale of your workstation.
- [APTCustom](#)
Use the APTCustom actions to customize your APT applications.
- [ConcatLineValues](#)
Use the ConcatLineValues actions to merge line items and page fields in your APT applications.
- [Documents](#)
Use the Documents actions to create and manage documents on your APT applications.
- [FlexID](#)
Use the FlexID action to run the panel for the FlexID task.
- [Intellocate_Learning](#)
Use the Intellocate_Learning actions to allow APT to determine the new zones in your invoice image and add these zones to the DCO.
- [PageID](#)
Use the PageID actions to work with pages in your APT documents.
- [PreVerifySetup](#)
Use the PreVerifySetup action to set the label variable before you run the Verification task on the batch of documents.
- [Redaction](#)
Use the Redaction actions to redact selected information in your APT invoice images for security purposes.

Parent topic: [Application specific actions](#)

APT_Localization

Use the APT_Localization actions to work with currency values when you process documents generated from a locale that is different from the locale of your workstation.

The APT_Localization actions are described in the following table.

Action	Description
--------	-------------

Action	Description
CheckAndFixLocalDecimal	Checks and fixes issues when the decimal in a currency value is not recognized and is replaced with a comma or a space. Also converts the decimal separator to use the local setting.
ConvertToLocalDecimal	Reads the computer settings and sets the decimal place.
IsFieldLocalCurrency	Queries the local decimal separator and checks to see whether the field contains a local currency value.
IsLocalDecimalSeparator	Checks to see whether the decimal separator on the workstation matches the locale of the document that you are processing.
IsOriginalEuroFormat	Checks the document level Euro Format and USCanUK Format variables for the currency values in the recognized data. Votes on the decimal format that is used in the document.
IsWorkstationLocale	Checks to see which decimal separator is used on the workstation.

Parent topic: [Datacap Accounts Payable actions](#)

APTCustom

Use the APTCustom actions to customize your APT applications.

The APTCustom actions are described in the following table.

Action	Description
AddAllTaxesToTaxField	Takes every tax detail line value and adds it to a field called Tax.
AddToDate	Specifies a time to add to the date value stored in a field.
CalculateInvoiceTotalLocalized	Calculates the total value of the localized invoice.
CalculateLineItemLocalized	Calculates the value of the localized line item.
ClearCurrentField	Erases the data from the current field.
ConvertEuroDateToUS	Converts the Date values from a European format to a US format.
ConvertUSDateToEuro	Converts the Date values from a US format to a European format.
FindTaxValue	Locates the taxes on an invoice.
IsDate_FormatEuro	Checks to see whether the Date value is in a valid European format.
IsInvoiceFromUS	Checks to see whether the invoice is from a company in the United States.
MakeFieldHighConfidence	Sets all of the character confidence field values to high confidence.
PopulateTaxType	Populates the Tax Details section with Tax Type information.

Parent topic: [Datacap Accounts Payable actions](#)

ConcatLineValues

Use the ConcatLineValues actions to merge line items and page fields in your APT applications.

The ConcatLineValues are described in the following table.

Action	Description
MergeLineItemFieldToPageField	Merges the line item values on the invoice image into a page variable
MergePageFieldToDocVar	Merges the page field values on the invoice images into a document variable

Parent topic: [Datacap Accounts Payable actions](#)

Documents

Use the Documents actions to create and manage documents on your APT applications.

The Documents actions are described in the following table.

Action	Description
CombinePreviousDoc	Copies the pages of the previous named document into the front of this document.
CountPagesToDocVar	Counts the number of page objects in a document and writes the result in a document variable.
IsFirstDocInBatch	Checks to see whether this object is the first document in the batch.
RemoveDocumentStructure	Levels the document/page hierarchy from the batch. For example, if the batch consists of multiple documents that each contain a set of pages, the document level is removed. All of the pages become a flat structure.

Parent topic: [Datacap Accounts Payable actions](#)

FlexID

Use the FlexID action to run the panel for the FlexID task.

The FlexID action is described in the following table.

Action	Description
RunFlexIDPanel	Runs the panel for the FlexID task in APT.

Parent topic: [Datacap Accounts Payable actions](#)

Intellocate_Learning

Use the Intellocate_Learning actions to allow APT to determine the new zones in your invoice image and add these zones to the DCO.

The Intellocate_Learning actions are described in the following table.

Action	Description
Learn_Zones	Learns about a new zone and adds the zone to the DCO.

Action	Description
Learn_ZonesFPX	Learns about a new zone and adds the zone to the DCO. Learn_ZonesFPX is FPXML compatible and it learns only the first line item in the zone. To write the zone to FPXML, this action must be followed by a WriteZonesFPX action.

Parent topic: [Datacap Accounts Payable actions](#)

PageID

Use the PageID actions to work with pages in your APT documents.

The PageID actions are described in the following table.

Action	Description
PageIDByBCSep	Separates the pages in the batch by using bar code separator sheets.
PageIDBySeqTypes	Applies a sequence of comma-separated page types, in order, to the pages of type Other in a batch, and repeats the sequence indefinitely.
PageIDByVariableChange	Processes all of the pages with a page type that is set according to the input parameters. For example, when pages are PageID, a variable for the page can be watched so when that variable changes, a new page type is created.

Parent topic: [Datacap Accounts Payable actions](#)

PreVerifySetup

Use the PreVerifySetup action to set the label variable before you run the Verification task on the batch of documents.

The PreVerifySetup action is described in the following table.

Action	Description
SetLabels	Reads the .ini file and sets up label variable that is used by Datacap Web Client and Datacap Desktop to display the label on each field.

Parent topic: [Datacap Accounts Payable actions](#)

Redaction

Use the Redaction actions to redact selected information in your APT invoice images for security purposes.

The Redaction actions are described in the following table.

Action	Description
EraseRect	Redacts the area of the invoice image that you specify in pixels. For example, you can configure this action as EraseRect(x1, y1, y2, bBlack) to redact the area in the image that corresponds to the specified coordinates.

Action	Description
GetAll Barcodes	Gets as many as 10 bar codes from the invoice image.
RedactByRegex	Checks for a specified regular expression and redacts all occurrences of that regular expression.
RedactField	Redacts an area of the invoice image by using the current coordinates of the field.

Parent topic: [Datacap Accounts Payable actions](#)

Adding keyboard shortcuts to an application

You can create keyboard shortcuts for navigating and selecting user-interface controls within a specific task in an application. To add a new shortcut to an application, you must edit the corresponding task's configuration XML file.

About this task

Any existing keyboard shortcuts, which are defined by the default Datacap installation, take precedence over new keyboard shortcuts. When you add shortcuts, ensure that they do not conflict with existing shortcuts.

Procedure

To add a keyboard shortcut for a task in an application:

1. Open the task's configuration XML file.
Tip: The task's configuration XML file is located at *default installation location\application name\dco_application name\task name.set.xml*. For example, the location of the configuration file for the Fixup task in the TravelDocs application is C:\Datacap\TravelDocs\dco_TravelDocs\fixup.set.xml.
 2. Determine the user-interface control for which you want to add a keyboard shortcut. For example, to add a shortcut to a control that is named Page Status, search for `label="Page Status"` in the XML file.
 3. On the same line where you located the `label`, add this code: `hotkey=`.
 4. Enter a value for `hotkey=`. For example, if you enter `hotkey=A`, then Ctrl-A is the keyboard shortcut for selecting Page Status.
 5. Save the XML file.
- [Keyboard shortcuts for the ASPX web pages](#)
You can use the Datacap Web Client keyboard shortcuts to view images on the aVerify, ImgEnter, Restruct, and VeriFine web pages.
 - [Keyboard shortcuts for the Datacap Desktop Scan task](#)
You can use keyboard shortcuts to navigate and select user-interface controls in Datacap Desktop.
 - [Keyboard shortcuts for the Datacap Desktop Fixup task](#)
You can use keyboard shortcuts to navigate and select user-interface controls in Datacap Desktop.
 - [Keyboard shortcuts for the Datacap Desktop Verify task](#)
You can use keyboard shortcuts to navigate and select user-interface controls in Datacap Desktop.

Keyboard shortcuts for the ASPX web pages

You can use the Datacap Web Client keyboard shortcuts to view images on the aVerify, ImgEnter, Restruct, and VeriFine web pages.

Table 1. Keyboard shortcuts for the aVerify.aspx page

To Do This	Do This
Go to the next image in a multipage document	Shift+Alt, then press >
Go to the previous image in a multipage document	Shift+Alt, then press <
Zoom in image (image expands)	Shift+Alt, then press +
Zoom out image (image shrinks)	Shift+Alt, then press -
Zoom image by quarter	Shift+Alt, then press Q
Scroll image	Shift+Alt, then press Up Arrow, Down Arrow, Left Arrow, or Right Arrow

Table 2. Keyboard shortcuts for the ImgEnter.aspx page

To Do This	Do This
Next LC and Problem field	Alt+I
Go to next field in DCO order by using the Tab key	Press Tab, then press Enter
Go to previous field in DCO order by using the Tab key	Shift+Tab, then press Enter
Change the orientation of the edit box to be displayed either above or below a snippet	Shift+Up Arrow or Shift+ Down Arrow
Shift all field zones (offset fields)	Ctrl+Shift+Arrows
Move field rectangle to change the field position variable	Alt+Shift+Arrows
Resize current field zone	Alt+Arrows
Select all in the field	Ctrl+A
Clear all in the field	Ctrl+X
Navigate to the start of the field value and to the end of the field value	PageUp, PageDn

Table 3. Keyboard shortcuts for the Restruct.aspx page

To Do This	Do This
Select next node in tree structure	Press Down Arrow
Select previous node in tree structure	Press Up Arrow
Move page down or up in tree structure	Shift+Down Arrow or Shift+Up Arrow
Collapse node to hide child in tree structure	Ctrl+Left Arrow
Expand node to show child in tree structure	Ctrl+Right Arrow
Split document in tree structure	Shift+Right Arrow

Table 4. Keyboard shortcuts for the VeriFine.aspx page

To Do This	Do This
Next LC	Alt + L
Load the next or previous image in a multipage document	Shift+Alt, then press Page Up or Page Down
Select next node in the tree structure	Press Down Arrow
Select previous node in the tree structure	Press Up Arrow
Move page down or up in the tree structure	Shift+Down Arrow or Shift+Up Arrow
Collapse node to hide child in the tree structure	Ctrl+Left Arrow
Expand node to show child in the tree structure	Ctrl+Right Arrow
Split document in the tree structure	Shift+Right Arrow

Parent topic: [Adding keyboard shortcuts to an application](#)

Keyboard shortcuts for the Datacap Desktop Scan task

You can use keyboard shortcuts to navigate and select user-interface controls in Datacap Desktop.

Depending on how the corresponding Datacap Desktop task is configured, some of the keyboard shortcuts might not be applicable in your environment.

Table 1. General keyboard shortcuts for the Datacap Desktop Scan task

Keyboard shortcut	Function
Alt + E	Open Edit menu
Alt + N	Open Navigate menu
Alt + I	Open View menu
Alt + M	Open Image menu
Alt + H	Open Help menu
F6	Submit, Done, or Finish
F7	Full image view
F8	Fit image to width
Ctrl + Q	Place batch on Hold
Alt + Q	Quarter Zoom displays sections of the image.
Ctrl + Alt + I	Toggle the Image View
Ctrl + Alt + S	Toggle the Batch View
Alt + I, then press S, then press Enter	Displays the Start Batch Panel.

Table 2. Scan Settings keyboard shortcuts

Keyboard shortcut	Function
-------------------	----------

Keyboard shortcut	Function
Alt + S	Open Scan Settings menu
Alt + Shift + B	Save as batch settings
Alt + Shift + U	Save as user settings
Alt + Shift + S	Save as station settings
Alt + Shift + C	Save as custom settings
Ctrl + Shift + C	Load custom settings
Ctrl + Shift + B	Load batch settings
Ctrl + Shift + U	Load user settings
Ctrl + Shift + S	Load station settings

Parent topic: [Adding keyboard shortcuts to an application](#)

Keyboard shortcuts for the Datacap Desktop Fixup task

You can use keyboard shortcuts to navigate and select user-interface controls in Datacap Desktop.

Depending on how the corresponding Datacap Desktop task is configured, some of the keyboard shortcuts might not be applicable in your environment.

Table 1. Datacap Desktop Fixup keyboard shortcuts

Keyboard shortcut	Function
F6	Finish batch
F5	Refresh
Delete	Remove image
Insert	Insert scanned images
Ctrl + Z	Cancel batch
Ctrl + B	Focus on Batch Tree
Ctrl + C	Stop Scanning
Ctrl + D	Delete (Remove)
Ctrl + F	Rotate image 180°
Ctrl + L	Rotate Left (90° counterclockwise)
Ctrl + O	Open
Ctrl + P	Replace (rescan)
Ctrl + R	Rotate Right (90° clockwise)
Ctrl + S	Focus on Scan Panel
Ctrl + T	Toggle thumbnail view
Ctrl + U	Navigate backwards through images. (Manual image selection mode only. For more information, see Configuring the image selection mode for the Datacap Desktop Fixup task).

Keyboard shortcut	Function
Ctrl + Y	Navigate forwards through images. (Manual image selection mode only. For more information, see Configuring the image selection mode for the Datacap Desktop Fixup task).
Ctrl + F8	Display the page type
Ctrl + F9	Mark the page for review
Ctrl + Up Arrow	Move up
Ctrl + Down Arrow	Move down

Parent topic: [Adding keyboard shortcuts to an application](#)

Keyboard shortcuts for the Datacap Desktop Verify task

You can use keyboard shortcuts to navigate and select user-interface controls in Datacap Desktop.

Depending on how the corresponding Datacap Desktop task is configured, some of the keyboard shortcuts might not be applicable in your environment.

Table 1. Keyboard shortcuts for Datacap Desktop

Keyboard shortcut	Menu	Command
Ctrl + P	Navigate	Go to Previous Problem Page
Ctrl + N	Navigate	Go to Next Problem Page
Ctrl + Shift + P	Navigate	Go to Previous Page
Ctrl + Shift + N	Navigate	Go to Next Page
Alt + L	Navigate	Go to Next Low Confidence (LC) Field
Alt + V	Navigate	Validate
F6	Navigate	Submit
Ctrl + Q	Navigate	Place batch on Hold
F11	Edit	Clear All Fields
Ctrl + Shift + A	Edit	Automatic login
Ctrl + Alt + I	View	Toggle the Image View
Ctrl + Alt + S	View	Toggle the Batch View
Alt + Shift + P	View	Display Previous Image in document
Alt + Shift + N	View	Display Next Image in document
Alt + Shift + NumPad8	Snippet	Move up
Alt + Shift + NumPad2	Snippet	Move down
Alt + Shift + NumPad4	Snippet	Move left
Alt + Shift + NumPad6	Snippet	Move right
Alt + Shift + I	Snippet	Zoom in
Alt + Shift + O	Snippet	Zoom out

Keyboard shortcut	Menu	Command
Alt + Shift + R	Snippet	Restore
F4	Image	Draw Fields
Alt + Q	Image	Zoom Quarter displays sections of the image.
F8	Image	Fit to width
F7	Image	Full image view
Alt + W	Image	Draw CCO Words
Alt + R	Image	Draw CCO Lines
Alt + Shift + 9	Image	Rotate 90°
Alt + Shift + F	Image	Rotate 180°
Alt + Shift + 7	Image	Rotate 270°

Parent topic: [Adding keyboard shortcuts to an application](#)

Główny indeks dokumentacji Datacap

[Znaki specjalne](#)

[Znaki liczbowe](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Z](#)

Znaki specjalne

- plik .app aplikacji
 - [Application Manager](#)
- plik .app
 - [zapisywanie haseł](#)
- [@APPPATH\(<ścieżka_kluczowa>\)](#)
- [@APPVAR\(<ścieżka_kluczowa>\)](#)
- [@B\](#)
- [@BATCHID](#)
- [@CHR\(<wartość_Unicode>\)](#)
- [@D\](#)
- [@DATE\(<format>\)](#)
- [@DCO\(<nazwa_właściwości>\)](#)
- [@DICT_VALUE\(<pole>\)](#)
- [@DICT_VINDEX\(<łańcuch_csv>\)](#)
- [@DICT_WINDEX\(<łańcuch_csv>\)](#)
- [@DICT_WORD\(<pole>\)](#)
- [@EMPTY](#)
- [@F\](#)
- [@F.<nazwa_zmiennej>](#)
- [@ID](#)
- [@JOBID](#)
- [@JOBNAME](#)
- [@LOCALE](#)
- [@OPERATOR](#)
- [@P\](#)

- @P.<nazwa_zmiennej>
- @PATH(<klucz>)
- @PILOT(<nazwa_właściwości>)
- @PROCESSDIR
- @PROJECTDIR
- @STATION
- @STATUS
- @STRING(<wartość_łańcuchowa>)
- @TASKID
- @TASKNAME
- @TIME(<format>)
- @TYPE
- @VALUE
- @VAR(<nazwa_zmiennej>)

Znaki liczbowe

- 4010Common
- 4010Institutional
- 4010Professional
- 5010, konfiguracja formularza dla instytucji
- 5010, konfiguracja formularza firmowego
- 5010Common
- 5010Institutional
- 5010Professional

A

- AbortOnError, działanie (1), (2)
- dostęp do ustawień konfiguracji aplikacji
- dostęp do innych informacji
- prawa dostępu
 - przypisywanie do baz danych
 - baza danych
- dostęp do informacji o czynności (1), (2)
- dostęp do hierarchii środowisk wykonawczych (1), (2), (3), (4)
- Uzyskiwanie dostępu do obiektu SetupNode
 - DCO, metody
- Uprawnienia dostępu do konta
 - c:\Datacap
 - c:\Datacap\application\batches
 - c:\Datacap\application\fingerpint
 - Datacap Maintenance Manager (1), (2), (3), (4)
 - Datacap\RRS, folder
 - Fingerprint, usługa (1), (2)
- ustawienia konta
 - Datacap Maintenance Manager
- uprawnienia udostępniania konta
 - Datacap Maintenance Manager
- konta
 - tworzenie dla Report Viewer
 - tworzenie dla Rulerunner
 - Datacap
 - usuwanie

- Fingerprint Service
 - usuwanie Datacap
- szczegóły działania
 - szczegóły
 - przeglądanie
- podsumowania bibliotek działań
- działania
 - specyficzna dla aplikacji
 - APT_Localization
 - APTCustom
 - Autodoc
 - BlankPagesIDBySize
 - CalculateOffset
 - CreateFingerprint
 - DeleteFingerprint
 - FindBlackFingerprint
 - FindFingerprint
 - FindTemplate
 - MergeCCOs_ByType
 - SetApplicationID
 - SetFilter_HostName
 - SetFilter_PageType
 - SetFingerprint (1), (2)
 - SetFingerprint FailureThreshold
 - SetFingerprint WebserviceURL
 - SetFingerprintSearchArea
 - SetMaxOffset
 - SetProblemValue
 - SetSearchArea
 - SetTemplateDir
 - UpdateFingerprintStats
 - Kod kreskowy_P
 - Get2DCodeBP
 - GetAllBarcodesBP
 - GetBarcodeBP
 - GetDataMatrixCodeBP
 - IdentifyByBarcodesBP
 - MatchBarcodeBP
 - MatchBarcodePrefixBP
 - ReadBarCodeBP
 - SetMinimumConfidenceBP
 - Kod kreskowy_X
 - GetBarCode
 - MatchBarcode
 - ReadBarCode
 - CC
 - ClassifyCC
 - ClassifyTextCC
 - FindFingerprintCC
 - RunDecisionPlanCC
 - RunDecisionPlanForBlocksCC
 - RunDecisionPlanForTextCC
 - SetDecisionPlanCC

- SetDecisionPlanFieldsCC
 - SetKnowledgeBaseCC
 - SetLanguageCC
 - SetListenerURLCC
 - SetProblemValueCC
 - UpdateKnowledgeBaseCC
- Cco2cco
 - NormalizeCCO
 - SetMaxCharacter HeightAVG
 - SetMaxCharacter HeightTMM
- ClassifyLayout
 - Opinia
 - Identyfikacja
- CMISClient
 - CMISCreateFolder
 - CMISDeleteFile
 - CMISDeleteFolder
 - CMISDoesFileExist
 - CMISDoesFolderExist
 - CMISDownloadFile
 - CMISLogDocumentTypes
 - CMISLogin
 - CMISRefreshClientCache
 - CMISSetDocUploadProperty
 - CMISSetDocUploadType
 - CMISSetVersion
 - CMISUploadFile
 - CMISUploadPage
- ColorToBW
 - Przekształcenie_C2BW
 - C2BW_SetAttributes
- ConcatLineValues
- Konektory
- Przekształcanie
 - ExcelAutoFitColumns
 - ExcelAutoFitRows
 - ExcelOrientation ToLandscape
 - ExcelOrientationToPortrait
 - ExcelPrintBlankPage
 - ExcelPrintGridlines
 - ExcelPrintQuality
 - ExcelScalingFactor
 - ExcelTiffCompression
 - ExcelWorkbookToImage
 - ExcelWorkbookToImageEx
 - ExceptionSetFileTypes
 - ExceptionSetHandler
 - ExceptionSetTaskCondition
 - ExceptionSetVariableName
 - HtmlLayout
 - HtmlPrintQuality
 - HtmlTiffCompression
 - HtmlToImage

- ImageDefaultDPI
- ImageFileTypesToConvert
- ImageMonoThreshold
- ImageMonoType
- ImageToTIFF
- OutlookMessageTo AttachmentOnly
- OutlookMessageToImage AndAttachment
- OutlookPrintQuality
- OutlookTiffCompression
- PDFBitDepth
- PDFCompression
- PDFConversionMethod
- PDFConversionMode
- PDFDocumentToImage (1), (2)
- PDFGrayscale
- PDFHorizontalResolution
- PDFImageCompression
- PDFImageFileExtension
- PDFImageFileResolution
- PDFImageUse FastBinarization
- PDFJpegQuality
- PDFQuality
- PDFVerticalResolution
- RtfPrintQuality
- RtfTiffCompression
- RtfToImage
- SetNamePattern
- SplitMultipageTiff
- SplitTIFFCompression
- TxtFontName
- TxtFontSize
- TxtPrintQuality
- TxtTiffCompression
- TxtToImage
- WordDocumentToImage
- WordDocumentToPdf
- WordMonochromeQuality
- WordPrintQuality
- WordTiffCompression
- ZipOverwrite
- ZipPassword
- ZipUnPack
- Datacap Accounts Payable
- DatacapBOX
 - AddParentDataToPageMetadata
 - BackupFolder
 - CreateBatchSubfolder
 - DCOVarsAreMetadata
 - DocumentsToPDF
 - Pobieranie
 - FailIfFileExists
 - FieldsAreMetadata
 - ImportAsDocumentType

- ImportLimit
 - LookforExtensions
 - OverwriteExistingFiles
 - ProcessChildren
 - ReplaceMetadata
 - SourceFolder
 - TargetFolder
 - Przesyłanie
- dci_clipfield
- DCImageFix
 - ImageEnhance
 - LoadSettings
 - LoadSettings_FingerprintID
- DCO
 - ChkConfidence
 - ChkDCOStatus
 - ChkDCOType
 - ChkIntegrity
 - ChkLastDCOType
 - ClearAltText
 - ClearDCO
 - CopyPD2DD
 - CountPagesToDocumentVar
 - CreateDocuments
 - CreateFields
 - DeleteFields
 - IsDocumentCountMoreThan
 - IsFirstDocumentInBatch
 - JoinPreviousDocument
 - PropagateToAltText
 - RemoveDocumentStructure
 - SetDCOStatus
 - SetDCOType
 - SetDocStatus
 - SetDocumentType
 - SetFldConfidence
 - SetPageFingerprintID
 - SetPageStatus
 - SetPageTemplateID
 - SetPageType
- dcpdf
 - dcpdf_CreateTiffFrom PDF_CreateDocs
 - dcpdf_CreateTiffFromPDF
 - dcpdf_MakePDFDoc
 - dcpdf_MaxSizeToReconvert
 - dcpdf_SetApplication
 - dcpdf_SetAuthor
 - dcpdf_SetImage Compression
 - dcpdf_SetImageBitcount
 - dcpdf_SetImageGrayscale
 - dcpdf_SetImageQuality
 - dcpdf_SetImageResolution
 - dcpdf_SetKeywords

- dcpdf_SetProducer
 - dcpdf_SetSubject
 - dcpdf_SetTitle
 - dcpdf_UseAltConversion Method
- DocumentAnalytics, działania
 - AnalyzeLayout
 - CopyAllBlocks
 - CopyLabelValuePairs
 - CreateCcoFromLayout
 - CreateHTML
 - ExtractText
 - ExtractTextAlchemyLanguage
 - ExtractTextLogEnable
 - FindExtractedText
 - FindLabelValuePair
 - FindLabelValuePairs
 - FindPatterns
- Dokumenty
- Documentum
 - DM_Logon
 - DM_SetContentType
 - DM_SetFolderName
 - DM_SetObjectName
 - DM_UploadDocument
 - DM_UploadPage
- E-mail
 - SendEMail
 - SetAttachment
 - SetBlindCarbonCopyRcpts
 - SetCarbonCopyRcpts
 - SetEmailBody
 - SetMailServer
 - SetRecipients
 - SetSender
 - SetSubject
- Wyrównywanie
 - EqualizeUnbalancedImage
- Ewsmail
 - czas_przerwania_eksportu
 - gotowy_folder_wyeksportowany
 - eksport_EMLOption
 - wersja_ews_eksportu
 - limit_czas_HTTP_eksportu
 - opcja_właściwości_przesyłania_eksportu
 - logowanie_do_eksportu
 - wylogowywanie_z_eksportu
 - ex_max_docs
 - folder_probleków_eksportu
 - skan_eksportu
 - typy_eksportu
 - czas_oczekiwania_na_eksport
- Eksportowanie
 - BatchVariable_ExportValue

- BlankFields
- BlankLines
- BPilot
- CloseExportFile
- DCOProperty
- DocumentVariable_Export Value
- ExportAllFields
- ExportFieldValue
- ExportMYValue
- ExportSmartParameter
- ExportToBatchDir
- Filler
- FixedLenLJ
- FixedLenRJ
- GetDATE
- GetProfileString
- GetTime
- LineItem_AddElement
- LineItem_BlankFields
- LineItem_ClearElements
- LineItem_ExportElements
- LineItem_SmartParameter
- NewLine
- PageVariable_ExportValue
- ResetFieldVariables
- SaveFilePathAsVariable
- SetCSV
- SetElementSeparator
- SetExportPath
- SetExtensionName
- SetFileName
- SetFill
- SetFixedLength
- SetIgnoreFieldStatus
- SetJustified
- SetOMR_Separator
- SetSpaceFill
- SetZeroFill
- Tekst
- Variable_ExportValue
- Variable_IsValue
- ExportDB
 - AddRecord
 - ExportBatchIDToColumn
 - ExportCloseConnection
 - ExportFieldToColumn
 - ExportNodeXMLToColumn
 - ExportOpenConnection
 - ExportPropertyToColumn
 - ExportSmartParam ToColumn
 - ExportToColumn
 - SetTableName
- ExportXML

- xml_CommitNode
 - xml_NewNode
 - xml_SaveFile
 - xml_SetAttributeValue
 - xml_SetExportPath
 - xml_SetFileName
 - xml_SetNodeValue
- Faks
- FileIO
 - CheckFreeDiskSpace
 - CopyDirectory
 - CopyFile
 - DeleteDirectory
 - DeleteFile
 - GetFileSize
 - GetProfileString
 - IsDirectoryPresent
 - IsFilePresent
 - IsFileReadOnly
 - IsProfilePresent
 - RenameFile
 - SetFileReadOnly
 - SetProfileString
 - SplitFileName
- FileNet P8
 - ExcelWorkbookToPdf
 - FNP8_CreateFolder
 - Logowanie_FNP8
 - FNP8_MultiPageDocs
 - FNP8_SetDestinationFolder
 - FNP8_SetDocClassId
 - FNP8_SetDocTitle
 - FNP8_SetFileMimeType
 - FNP8_SetFileType
 - FNP8_SetKeyProperty
 - FNP8_SetLocale
 - FNP8_SetMultiValue Property
 - FNP8_SetProperty
 - FNP8_SetPropertyEx
 - FNP8_SetRetry
 - FNP8_SetTargetClassID
 - FNP8_SetTargetObjectID
 - FNP8_SetTimeout
 - FNP8_SetUploadMode
 - FNP8_SetURL
 - FNP8_UpdateProperties
 - Przesyłanie_FNP8
 - FNP8_UploadDir
 - HtmlToPdf
 - standardy zgodności PDF
 - typy kompresji obrazów PDF
 - PDFFREDocumentToImage
 - RtfToPdf

- TxtToPdf
- FileNetIDM
 - AddAllImagesToDocument
 - AddFileToDocument
 - AddPDFImageToDocument
 - AddTIFFImageToDocument
 - CreateFolder
 - FileNetDB_ADOConnect
 - FileNETDocID_SaveAs SmartParameter
 - FileNETDocID_SetValue
 - GetDocuments
 - GetTopFolders
 - IndexProperty_ID_DateComponent
 - IndexProperty_ID_Component
 - IndexProperty_LeftJUSTIFY
 - IndexProperty_RightJUSTIFY
 - IndexProperty_SmartParameter
 - IndexProperty_ID_Value
 - Library_DMA_Initialize
 - Library_DS_Initialize
 - Library_IS_Initialize
 - Library_LogIn
 - Library_LogOff
 - NewDocument
 - SaveDocToFolder
 - Przesyłanie
 - Upload_SetNumAttempts
 - UseIndexes_OFF
 - UseIndexes_ON
- FingerprintMaintenance
 - CloseDatabase
 - DeleteFingerprint
 - DeleteFingerprints
 - OpenDatabase
 - SetFingerprintFolder
- FlexID
- FPXML
 - ReadZonesFPX
 - SetDetailsAndLineitem PairFPX
 - SetDirectoryFPX
 - WriteZoneFPX
 - WriteZonesFPX
- globalny
- Skala szarości
 - ConvertGraytoBW
- IBMCM
 - IBMCM_SetChildAttributeValue
 - IBMCM_AddPages
 - IBMCM_CreateChildItem
 - IBMCM_CreateFolder
 - IBMCM_CreateItem
 - IBMCM_DeletePages
 - Logowanie_IBMCM

- IBMCM_ReplacePage
 - IBMCM_SearchItem
 - IBMCM_SetAttributeValue
 - IBMCM_SetDestinationFolder
 - IBMCM_SetMimeType
 - IBMCM_StoreItemIDinDCO
 - IBMCM_UploadDCO_DOC
 - IBMCM_UploadDCO_Page
- ICR_C
 - EnableLoggingICR_C
 - RecognizeFieldICR_C
 - RecognizeFieldVoteICR_C
 - RecognizePageFields ICR_CEx
 - RecognizePageFields2 CCO_ICR_C
 - RecognizePageFieldsICR_C
 - RecognizePageICR_C
 - RecognizePageToPDFICR_C
- ICR_P
 - AddWord
 - DeleteWord
 - ImportCSF
 - LoadFromFile
 - NewDictionary
 - RecognizeFieldsICR_P
 - SaveToFile
 - SetPostalDBPathICR_P
- ImageConvert
 - AppendAllImages
 - AppendAllImages_ByType
 - AppendImage
 - AppendImage_StartAsNew
 - ConvertToJPEG
 - ConvertToTIFF
 - SetChrominanceFactor
 - SetDeleteOriginal
 - SetGrayScale
 - SetLuminanceFactor
 - SetTIFFCompression
- ImageFix
- Imail
 - czas_przerwania_importu
 - import_AcceptMixedAttachments
 - import_AcceptNoAttachments
 - gotowy_folder_importu
 - logowanie_do_importu
 - wylogowanie_z_importu
 - im_max_docs
 - folder_probleków_importu
 - skan_importu
 - import_SetProxy
 - import_SortByDate
 - import_StoreEML
 - typy_importu

- import_UseSSL
 - czas_oczekiwania_na_import
- Imprint
 - AnnotateImage
 - ImPrint
 - Redact
 - RedactByRegEx
 - RedactParameters
 - SetAdjustedWidth
 - SetFontName
 - SetFontSize
 - SetOpaque
- zawieranie w aplikacjach
- instalacja
 - weryfikowanie
- Intellocate
 - iloc_AdjustZones
 - iloc_AssignPageType
 - iloc_SetDetailZones
 - iloc_SetZones
 - IsPageDataMissing
- Intellocate_Learning
- Faktura
 - AddToDetailErrorMsg
 - AddToErrorMsg
 - AllMixedCase
 - AllowOnlyChars
 - AlterDatebyDay
 - CalculateNotesZone
 - CaptureOpInfo
 - CheckAndFixDecimal
 - CheckForSticky
 - CheckFreeDiskSpace
 - ClearErrorMsg
 - CreateFingerprint
 - DetailFix
 - DoMsgbox
 - ExecuteSQLBind
 - FindExportImage
 - FPXMLUsed
 - GenerateDetails
 - iloc_SetDetailSimple
 - IncrementBatchVar
 - Is_InCharSet
 - Is_JobName
 - Is_JobNamePrefix
 - IsChildFieldBlank
 - IsChildFieldValue
 - IsCurrentObjValue
 - IsCurrentObjVariable
 - IsFingerPrintClass
 - IsInINI
 - IsInList

- IsMultipageDocument
- IsSinglePageDocument
- IsStationIDSuffix
- IsTaskName
- LoadCCOFromField
- MovePDF
- OpenConnection
- ParseImageName
- PopulateZNLineItem FieldDynamic
- ReadFPXMLZones
- IOverlay
 - Overlay
 - SetBackgroundImage
 - SetDitheringBackground
 - SetHaloBackground
- Lokalizacja
 - AddKeyList
 - AggregateKeyList
 - DefaultValue
 - FilterIt
 - FindDBList
 - FindDBList_InZone
 - FindKeyList
 - FindKeyList_InZone
 - FindLastKeyList
 - FindLastKeyList_InZone
 - FindLastRegEx
 - FindLastRegEx_InZone
 - FindLastRegExList
 - FindLastRegExList_InZone
 - FindLastWord
 - FindLastWord_InZone
 - FindNextDBList
 - FindNextDBList_InZone
 - FindNextKeyList
 - FindNextKeyList_InZone
 - FindNextRegExList
 - FindNextRegExList_InZone
 - FindRegExList
 - FindRegExList_InZone
 - GetSelectedBlockType
 - GoAboveWord
 - GoBelowWord
 - GoDownLine
 - GoFirstLine
 - GoFirstWord
 - GoLastLine
 - GoLastWord
 - GoLeftWord
 - GoRightWord
 - GoSiblingBlockNext
 - GoSiblingBlockPrevious
 - GoUpLine

- GroupWords
- GroupWordsLEFT
- GroupWordsRIGHT
- IsAlpha
- IsCurrency
- IsDateValue
- IsNumber
- IsSelectedBlockType
- IsValue
- IsValue_RegEx
- LocatePositionRestore
- LocatePositionSave
- MaxLength
- MergeWordLF
- MergeWordRT
- MinLength
- RegExFind
- RegExFind_InBlock
- RegExFind_InZone
- RegExFindNext
- RegExFindNext_InZone
- RegExFindNext_InBlock
- ScanRT
- SelectParentBlock
- SelectParentBlockOuterType
- SelectParentBlockType
- SelectSnippet
- SetRect
- UpdateDCOField
- UpdateField
- UpdateFieldWithBlock
- ValueInField
- ValueInField_Fuzzy
- ValueInField_RegEx
- WordFind
- WordFind_InZone
- WordFind_Offset
- WordFindNext
- WordFindNext_InZone
- Wyszukiwanie
 - ClearLookupResults
 - CloseConnection
 - ExecuteSQL
 - OpenConnection
 - PopulateWithResult
 - SmartSQL
- Maintenance Manager
 - LogClear
 - LogConfigure
 - LogSendEmail
 - LogWriteEventLog
 - LogWriteRecordSet
 - LogWriteSQLQuery

- ProcessChange BatchStatus
- ProcessChangeBatch StatusOrder
- ProcessChangeBatchStatus TaskOrder
- ProcessClearAuditTable
- ProcessClearDebugTable
- ProcessDeleteBatches
- ProcessDeleteBatchesEx
- ProcessInjectBatches
- ProcessMoveBatches
- ProcessMoveBatchesEx
- ProcessMoveDBRecords
- ProcessReset PendingOrNotify
- ProcessRunSqlQuery
- ProcessRunSqlQueryEx
- QueryClear
- QuerySetAge
- QuerySetBatchRange
- QuerySetBranch
- QuerySetDateFormat
- QuerySetDateRange
- QuerySetDateTimeFormat
- QuerySetGeneric
- QuerySetJobID
- QuerySetOperator
- QuerySetPriority
- QuerySetSeparator
- QuerySetStation
- QuerySetStatus
- QuerySetTaskID
- ReportQueryTMUsage
- ReportSetReportingTable
- ReportSetUsageDBTable
- SetAdminDB
- SetApplication
- SetEngineDB
- SetPassword
- SetServer
- SetStation
- SetupDisconnectAll
- SetupOpenApplication
- SetupOpenApplicationEx
- SetUser
- Identyfikowanie_MC
 - AutoField
 - FindFields
 - ReadDCOSetup
 - ReadPageSetup
 - SetFormType
 - SetMaxTolerantDistance
- Sprawdzanie poprawności_MC
 - AddCenturyTo2YearDigit
 - AddToDetailErrorMsg
 - AddToErrorMsg

- CalculateHCFALineCharges
- CalculateUBLLineCharges
- CheckDocID
- ClearErrorMsg
- CommonParseAddress
- CommonValAddress
- ConvertHyphen
- FilterPID
- FormatFieldLengths
- InheritSnippets
- MC_ReadZones
- Parse31aPhSig
- Parse58ainsnm
- Parse58binsnm
- Parse58cinsnm
- ParseConditionCodes
- ParseEPSDT
- ParseLastFirstIniNames
- ParseNDC
- PopulateFromField
- SetConf
- SetOriginalTIF
- StripTrailingAlpha
- TransformLI
- UpdateCredentialList
- ValidateNPI
- ValProcedureCode
- ValRequiredCode
- Medical Claims
- multi-pass verification
- mvscan
 - folder_zachowywania_mv
 - skanowanie
 - ustawienie_czasu_przerwania
 - ustawienie_folderu_kopiowania
 - ustawienie_usuwania_pustych_folderów
 - ustawienie_folderu
 - ustawienie_sprawdzenia_poprawności_obrazów
 - set_max_docs
 - ustawienie_typów_metadanych
 - ustawienie_wieku_minimalnego
 - ustawienie_czasu_ruchu_wstrzymania
 - ustawienie_serii_wielu_stron
 - ustawienie_folderu_problemów
 - ustawienie_metody_sortowania
 - ustawienie_trybu_drzewa
 - ustawienie_typów
 - ustawienie_czasu_oczekiwania
- OCR_A
 - EnableEngineLogsOCR_A
 - OCRA_ConvertImage2BW
 - Rozpoznawanie
 - RecognizeBarcodeOCR_A

- RecognizeFieldOCR_A
 - RecognizeFieldVoteOCR_A
 - RecognizePageFieldsOCR_A
 - RecognizePageOCR_A
 - RecognizeToALTOOCR_A
 - RecognizeToPDFOCR_A
 - ReleaseEngineOCR_A
 - RotateImageOCR_A
 - SetAutoRotationOCR_A
 - SetConfCalculationParamsOCR_A
 - SetFastModeOCR_A
- OCR_J
 - Rozpoznawanie
 - ReleaseEngine
- OCR_N
 - RecognizePageFieldsOCR_N
 - RecognizePageOCR_N
- OCR_S
 - RecognizeDocToPDF
 - RecognizeFieldOCR_S
 - RecognizeFieldVoteOCR_S
 - RecognizePageFields 2CCO_OCR_S
 - RecognizePageFieldsOCR_S
 - RecognizePageOCR_S
 - RecognizePageOCR_S_2TextFile
 - RecognizeToFile_OCR_S
 - RecognizeToPDF
 - RotateImage
 - SetEngineTimeout
 - SetFastTradeOffOCR_S
 - SetLegacyDecomposition OCR_S
 - SetOutOfProcessLoggingOCR_S
- OCR_SR
 - Rozpoznawanie
 - RecognizeFieldOCR_S
 - RecognizeFieldVoteOCR_S
 - RecognizePageFieldsOCR_S
 - RecognizePageOCR_S
 - RecognizeToFile_OCR_S
 - RecognizeToPDFOCR_S
 - RotateImage
 - RotateImageExOCR_S
 - SetEngineTimeoutOCR_S
- OcrRose
 - InitializeEngine
- OpenTextFaxServer
 - Łączenie
 - ContinueOn ConnectionError
 - ContinueOn FaxImportError
 - Rozłączanie
 - ImportFaxes
 - SendAsFax
 - SetAbortTimeout

- SetFaxRemovalAfterImport
 - SetInputFolder
 - SetMaxNumberOfFaxes
 - SetNumberOfRetries
 - SetPollingInterval
 - SetProcessedFaxesFolder
 - SetProtocol
 - SetRetryTimeout
 - SetServerName
 - SetUserID
 - SetUserPassword
 - SetWindowsAuthentication
- PageID
- PatternMatch
 - MatchPattern
 - pat_RecogMatch_Id
 - pat_RegisterZones
 - pat_ReleasePageAnchors
 - PatternMatch_Fingerprint
 - PatternMatch_Identify
 - PatternMatch_PageType
 - SetMatchConfidence
- Obraz
 - PIC_ApplyPictureString
 - PIC_FilterFields
 - PIC_FormatFields
 - PIC_ReplaceBlankField
 - PIC_SetPictureCharacter
 - PIC_ValidateField
- POLR
 - CallPOLR
- PreVerifySetup
- Recog_Shared
 - AnalyzeImage
 - CCONormalization_OFF
 - CreateTextFile
 - IsBlankPage
 - RecogContinueOnFailure
 - RecogOMRThresh
 - RecogOMRThreshold
 - RegisterPageFields
 - ReleaseImage
 - RotateTio
 - SetAdjustFieldToChars
 - SetFingerprintRecogPriority
 - SetFullPageRecogArea
 - SetOutOfProcess RecogTimeout
 - SetRecogFailureRetryDelay
 - SetupAutomaticRetry
 - SnapCCOtoDCO
 - SnapDCOtoCCO
 - SnapFieldtoChars
 - UseOutOfProcessRecog

- Redagowanie
- odwołanie do łańcuchów połączenia
- odwołanie do haset
- rrunner
 - AbortOnError
 - CheckAllIntegrity
 - CheckDocCount
 - CheckPageCount
 - DebugMode_OFF
 - DebugMode_ON
 - GoToNextFunction
 - MessageID
 - MessageIDParameter
 - PilotMessage_Clear
 - PilotMessage_Set
 - ProcessChildren
 - rr_AbortBatch
 - rr_Get
 - rr_WrireNode
 - rrAppend
 - rrCompare
 - rrCompareCase
 - rrCompareCaseLength
 - rrCompareNot
 - rrCompareNotCase
 - rrCompareNotCaseLength
 - rrCopy
 - rrPrepend
 - rrSet
 - SetBatchPriority
 - SetOperatorID
 - SetReturnValue
 - SetStationID
 - SetTaskStatus
 - SkipChildren
 - Zachowanie_statusu_WYŁĄCZONE
 - Zachowanie_statusu_WŁĄCZONE
 - Task_NumberOfSplits
 - Task_RaiseCondition
- SaveObjectVariable
- ScanLineItemDynamic
- SendOutlookNotification
- SetDynamicDetailZones
- SetPicChar
- SetStickyNo
- SetToDocIDMPTIFF
- SPExport
 - SP_CreateFolder
 - Logowanie_SP
 - SP_SetContentType
 - SP_SetFileType
 - SP_SetProperty
 - SP_SetUploadMode

- SP_SetUrl
 - Przesyłanie_SP
 - SP_UploadDir
- Podział
 - SplitBatch
- Statystyki
 - AddToDBTotals
 - CompareFieldsText
 - IsBatchAborted
 - SaveFieldsText
- SwapImages
- SwitchMMDD
- TifMerge
 - TifMerge_CheckStatus
 - TifMerge_ExportToBatchDir
 - TifMerge_MergeImages
 - TifMerge_MyImage
 - TifMerge_Preserve Compression
 - TifMerge_SetFileName
 - TifMerge_SetFilePath
- TM524
- TruncateFromStart
- UpdateFPStats
- ValidateVendor
- Sprawdzanie poprawności
 - AddLeadingZeros
 - AddPaddingToEnd
 - AddPaddingToLeft
 - AddPaddingToRight
 - AddPaddingToStart
 - AddTrailingZeros
 - AllowOnlyChars
 - AppendFromField
 - AppendToField
 - AssignFieldDefault
 - Obliczanie
 - CalculateDateDifference
 - CalculateFields
 - CheckSubFields
 - CompareFields
 - ConvertFieldTo Currency
 - ConvertToLowerCase
 - ConvertToUpperCase
 - CopyField
 - CopyFieldToField
 - DateStampField
 - DeleteAllAlpha
 - DeleteAllMiscChars
 - DeleteAllNumeric
 - DeleteAllPunct
 - DeleteAllSysChars
 - DeleteChildType
 - DeleteLCSpaces

- DeleteParentObj
- DeleteSelectedChars
- EmptyFieldValue
- FailRuleSet
- FieldContainsValue
- FilterFieldSelectedChars
- FormatNumberToLocale
- GetJobID
- HasChildOfType
- InsertChars
- InsertDecimalPoint
- IsFieldCurrency
- IsFieldDate
- IsFieldDateEqualOrAfter
- IsFieldDateEqualOrBefore
- IsFieldDateUpToToday
- IsFieldDateWithinRange
- IsFieldDateWithinXDays
- IsFieldDateWithReformat
- IsFieldEmpty
- IsFieldFilled
- IsFieldGreaterOrEqual
- IsFieldHidden
- IsFieldLengthMax
- IsFieldLengthMin
- IsFieldLessOrEqual
- IsFieldMatching
- IsFieldPercent NonNumeric
- IsFieldPercentAlpha
- IsFieldPercentNumeric
- IsMatchingJobID
- IsMaxOMRChecked
- IsMinOMRChecked
- IsPatternInField
- IsSupportedImageFile
- IsThisFieldEmpty
- IsThisFieldFilled
- IsVariableEmpty
- IsVariableFilled
- LeftTruncate
- MessageBox
- ParseMultilineAddress
- ParseName
- ReadCurrentObjVariable
- ReadFieldValue
- ReadPageVariableValue
- ReplaceChars
- ReplaceValueAtPosition
- ResetField
- RightTruncate
- SaveAsCurrentObjVariable
- SaveAsPageVariable
- SetIsOverrideable

- SplitFieldValueLeft
 - SplitFieldValuePreserveEnd (1), (2)
 - SplitFieldValueRight
 - SumFields
 - TimeStampField
 - TrimSpaces
 - TruncateFromEnd
- przeglądanie szczegółów
- Głosowanie
 - VoteFld
- Vscan
 - AddDocument
 - CopyFile
 - DeleteImageFile
 - MoveImageFileToDirectory
 - Skanowanie
 - SearchInSubdirectory
 - SetAlternateImageNames
 - SetFastMode
 - SetImageType
 - SetMailSourceFolder
 - SetMaxImageFiles
 - SetMultiPageTiff
 - SetSortOrder
 - SetSourceDirectory
- Usługi Web Services
 - WsClearHeaders
 - WsClearParameters
 - WsClearResultItems
 - WsEncodeParameter
 - WsGetFile
 - WsGetValues
 - WsSetCredentials
 - WsSetHeader
 - WsSetNamespace
 - WsSetParameter
 - WsSetResultItem
 - WsSetTimeout
 - WsUploadData
 - WsUploadFile
- WriteErrorMessage
- Strefy
 - AdjustZonesToImageOffset
 - AnchorPage
 - CalculateLocalOffset
 - CreateBlockCCO
 - FindBlocks_WhiteSpace
 - FindDataBlocks
 - FindLineItems
 - FindRegExBlocks
 - FindZoneLineItems
 - GetZoneText
 - InheritParentPosition

- [LoadBlockCCO](#)
 - [LoadZones](#)
 - [MCCOPositionAdjust](#)
 - [MergeZones](#)
 - [PadZone](#)
 - [PopulateZNField](#)
 - [PopulateZNLineItemField](#)
 - [ReadZones](#)
 - [RegisterPage](#)
 - [ScanDetails](#)
 - [ScanDetailsByLines](#)
 - [ScanDetailsByVSpace](#)
 - [ScanLineItem](#)
 - [SetEOL](#)
 - [SetEOL_CRLF](#)
 - [ZoneBOTTOM_ImageBottom](#)
 - [ZoneBOTTOM_LowerBound](#)
 - [ZoneBOTTOM_UpperBound](#)
 - [ZoneImage_SaveAs](#)
 - [ZoneLEFT_ImageLeft](#)
 - [ZoneLEFT_LeftBound](#)
 - [ZoneLEFT_RightBound](#)
 - [ZoneRIGHT_ImageRight](#)
 - [ZoneRIGHT_LeftBound](#)
 - [ZoneRIGHT_RightBound](#)
 - [ZoneTOP_ImageTop](#)
 - [ZoneTOP_LowerBound](#)
 - [ZoneTOP_UpperBound](#)
- Uwierzytelnianie Active Directory
 - [ADSI, uwierzytelnianie](#)
- dodawanie działań
 - [Datacap Maintenance Manager, aplikacje](#)
- Dodawanie listy aplikacji
 - Lista kontrolna
 - [Datacap.xml, plik](#)
- dodawanie odcisków
 - [biblioteka odcisków](#)
- Dodawanie zabezpieczeń
 - Lista kontrolna
 - [folder aplikacji Datacap](#)
 - [Datacap Web Client \(1\), \(2\)](#)
- dodawanie zaufanej strony
 - [Datacap Web Client](#)
- [AddAllImagesToDocument](#), działanie (1), (2)
- [AddAllTaxesToTaxField](#), działanie
 - [opis](#)
- [AddCenturyTo2YearDigit](#), działanie
- [AddChild](#)
 - [DCO, metody](#)
- [AddDocument](#), działanie (1), (2)
- [AddFileToDocument](#), działanie (1), (2)
- Dodawanie konta do grupy IIS_IUSRS
 - [Fingerprint, usługa](#)

- Dodawanie puli aplikacji
 - [Fingerprint, usługa](#)
- Dodawanie odwołań do aplikacji
 - Plik Datacap.xml [\(1\)](#), [\(2\)](#)
- dodawanie wartości danych dla znaków
 - [AddValue](#)
 - [DCO, metody](#)
- dodawanie obiektów podrzędnych
 - [AddChild](#)
 - [DCO, metody](#)
- [dodawanie odcisków](#)
- [dodawanie pojedynczo](#)
- Dodawanie zabezpieczeń
 - [Datacap Web Client](#)
 - [zaufana strona Opcje internetowe](#)
- Dodawanie użytkowników do Rulerunner
 - [ADLDS, uwierzytelnianie](#)
 - [LLLDAP, uwierzytelnianie](#)
- dodawanie zmiennych
 - [AddVariable](#)
 - [DCO, metody](#)
- [AddKeyList, działanie \(1\), \(2\)](#)
- [AddLeadingZeros, działanie \(1\), \(2\)](#)
- [AddNode](#)
 - [DCOSetup, metody](#)
- [AddPaddingToEnd, działanie \(1\), \(2\)](#)
- [AddPaddingToLeft, działanie \(1\), \(2\)](#)
- [AddPaddingToRight, działanie \(1\), \(2\)](#)
- [AddPaddingToStart, działanie \(1\), \(2\)](#)
- [AddParentDataToPageMetadata](#)
- [AddParentDataToPageMetadata, działanie](#)
- [AddPDFImageToDocument, działanie \(1\), \(2\)](#)
- [AddRecord, działanie \(1\), \(2\)](#)
- [AddRule](#)
 - [DCOSetupNode, metody](#)
- [AddTIFFImageToDocument, działanie \(1\), \(2\)](#)
- [AddToDate, działanie](#)
 - [opis](#)
- [AddToDBTotals, działanie](#)
- [AddToDetailErrorMsg, działanie \(1\), \(2\), \(3\)](#)
- [AddToErrorMsg, działanie \(1\), \(2\), \(3\)](#)
- [AddTrailingZeros, działanie \(1\), \(2\)](#)
- [AddValue](#)
 - [DCO, metody](#)
- [AddVariable](#)
 - [DCO, metody](#)
 - [DCOSetupNode, metody](#)
- [AddVariableFloat](#)
 - [DCO, metody](#)
- [AddVariableInt](#)
 - [DCO, metody](#)
- [AddVariableString](#)
 - [DCO, metody](#)

- AddWord, działanie (1), (2)
- AdjustZonesToImageOffset, działanie (1), (2)
- ADLDS, uwierzytelnianie
 - [Uwierzytelnianie użytkowników](#)
- administrowanie
 - [monitorowanie zadań](#)
- [administrowanie aplikacją](#)
- administrowanie
 - Datacap Navigator (1), (2)
- Administracja
 - Datacap Navigator (1), (2)
 - Datacap Web Client (1), (2)
- Administracyjna baza danych
 - [dostęp użytkownika](#)
- AdministrationDatacap Fingerprint Maintenance Tool
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
 - [Zarządzanie FastDoc](#)
 - [Zdalny dostęp](#)
 - [klient WWW, harmonogram przetwarzania](#)
- ADSI, uwierzytelnianie
 - [włączanie dla Report Viewer](#)
 - [Uwierzytelnianie grupy](#)
- AggregateKeyList, działanie (1), (2)
- agregaty
 - [dodawanie](#)
- [klient WWW AIndex](#)
 - [konfigurowanie](#)
 - [Statusy pól gotowych](#)
 - [Statusy stron gotowych](#)
 - [Statusy pól zignorowanych](#)
 - [ManualIDValidate, reguła](#) (1), (2)
 - [ustawienia ManualPageID](#)
 - [identyfikator strony](#)
 - [przekształcanie zadania wsadowego](#)
 - [aktualizowanie aplikacji](#)
 - [Statusy sprawdzania poprawności](#)
- AllMixedCase, działanie (1), (2)
- AllowOnlyChars, działanie (1), (2), (3), (4)
- AltConfidenceString
 - [DCO, właściwości](#)
- AlterDatebyDay, działanie (1), (2)
- AltText
 - [DCO, właściwości](#)
- AnalyzeImage, działanie (1), (2)
- [AnalyzeLayout, działanie](#)
- obiekty zakotwiczone
 - [dopasowywanie wzorca](#) (1), (2)
 - [konfigurowanie](#)
 - [używanie wielu zakotwiczeń](#)
- identyfikowanie wzorców zakotwiczenia
 - [dopasowywanie wzorca](#)
- AnchorPage, działanie (1), (2)

- [Android](#)
 - [automatyczny strefowy współczynnik OCR](#)
 - [klasyfikacja kodów kreskowych](#)
 - [geokodowanie i lokalizacja](#)
- [AnnotateImage, działanie \(1\), \(2\)](#)
- [API](#)
 - [tworzenie hierarchii dokumentów](#)
 - [DCO](#)
 - [Hierarchia dokumentów](#)
 - [Hierarchia zadań wsadowych Runtime](#)
 - [Konfiguracja DCO](#)
- [AppendAllImages, działanie \(1\), \(2\)](#)
- [AppendAllImages_ByType, działanie \(1\), \(2\)](#)
- [AppendFromField, działanie \(1\), \(2\)](#)
- [AppendImage, działanie \(1\), \(2\)](#)
- [AppendImage_StartAsNew, działanie \(1\), \(2\)](#)
- [AppendToField, działanie \(1\), \(2\)](#)
- [aplikacja](#)
 - [administracja](#)
 - [nawiązywanie połączenia z](#)
 - [debugowanie](#)
 - [aktualizowanie](#)
- [architektura aplikacji](#)
- [konfiguracja aplikacji](#)
 - [FastDoc](#)
 - [tryb Datacap Server](#)
- [pliki aplikacji](#)
 - [przenoszenie](#)
 - [zmiana nazwy](#)
- [Folder aplikacji](#)
 - [konfigurowanie zabezpieczeń](#)
- [zabezpieczenia folderów aplikacji](#)
 - [usługi Datacap Web Services](#)
- [Lista aplikacji](#)
 - [GET, metoda](#)
 - [GetApplicationList](#)
- [Application Manager](#)
 - [plik .app aplikacji](#)
 - [ADLDS, uwierzytelnianie](#)
 - [ADSI, uwierzytelnianie](#)
 - [Przypisywanie grupy do zadania wsadowego](#)
 - [Definiowanie niestandardowego filtra monitora zadań](#)
 - [Usługi Datacap Web Services \(1\), \(2\)](#)
 - [Konfigurowanie uwierzytelniania TMA](#)
 - [LDAP, uwierzytelnianie](#)
 - [LLLDAP, uwierzytelnianie](#)
 - [Zapisywanie, ustawienie aplikacji](#)
- [pule aplikacji](#)
 - [dodawanie dla Report Viewer \(1\), \(2\)](#)
- [Odwołania do aplikacji](#)
 - [Datacap.xml, plik](#)
- [ustawienia aplikacji](#)
 - [uzyskiwanie dostępu korzystając ze specjalnych zmiennych](#)

- Application Manager (1), (2), (3), (4), (5)
 - Włączanie zbioru statystyk aplikacji
 - ustawianie lokalizacji pliku datacap.xml (1), (2), (3), (4)
- Działania konfiguracyjne aplikacji
 - SetAdminDB (1), (2)
 - SetApplication (1), (2)
 - SetEngineDB (1), (2)
 - SetPassword (1), (2)
 - SetServer (1), (2)
 - SetStation (1), (2)
 - SetupDisconnectAll (1), (2)
 - SetupOpenApplication (1), (2)
 - SetupOpenApplicationEx (1), (2)
 - SetUser (1), (2)
- działania właściwe dla aplikacji
 - APT_Localization
 - APTCustom
 - ConcatLineValues
 - Datacap Accounts Payable
 - Dokumenty
 - FlexID
 - Intellocate_Learning
 - przegląd
 - PageID
 - PreVerifySetup
 - Redagowanie
- działania właściwe dla aplikacji
- zmienne właściwe dla aplikacji
- Kreator aplikacji
 - przekształcanie aplikacji (1), (2)
 - migracja (1), (2)
- aplikacje
 - architektura
 - opcje wiersza komend
 - konfigurowanie
 - konfigurowanie baz danych
 - DB2
 - Oracle
 - SQL Server
 - łańcuchy połączeń
 - przekształcanie z wcześniejszego wydania (1), (2), (3), (4), (5), (6), (7)
 - kopiowanie zadań wsadowych
 - kopiowanie na serwer Datacap
 - kopiowanie na serwer Datacap
 - dostosowywanie
 - Datacap
 - tworzenie
 - w kreatorze Application Wizard
 - Datacap Application Copy Tool
 - Datacap Maintenance Manager
 - dodawanie działań do
 - tworzenie (1), (2)
 - usuwanie zestawów reguł z

- używanie Datacap Studio
 - Datacap Web Client, przesyłanie, usługa
 - projektowanie
 - FastDoc
 - skróty klawiaturowe
 - lokalizacja
 - lokalizacja w sieci
 - Medical Claims (1), (2)
 - migrowanie (1), (2), (3), (4), (5)
 - przekształcanie paneli (1), (2), (3)
 - migrowanie od wcześniejszej wersji (1), (2), (3), (4), (5), (6), (7)
 - opcje migrowania
 - przenoszenie
 - przenoszenie do nowego środowiska
 - otwieranie
 - określanie formatu danych
 - rygorystyczne przesyłanie
 - tłumaczenie
 - aktualizacja w istniejącym środowisku
 - aktualizowanie (1), (2), (3), (4)
 - interfejs użytkownika
- APT
 - dodawanie elementów wiersza zamówienia
 - dodawanie dostawców do listy Demo Vendors
 - powiązanie dostawcy z odciskiem
 - przechwytywanie pól na obróconym obrazie
 - czyszczenie obrazów faktur (1), (2)
 - eksportowanie obrazów faktur (1), (2)
 - nadawanie uprawnień
 - obsługa nieznanymi faktur
 - identyfikowanie wierszy szczegółów na nieznanymi fakturach
 - logowanie do Datacap Web Client
 - wstępnie skonfigurowane zadania (1), (2)
 - przygotowywanie obrazów faktur
 - przetwarzanie obrazów faktur (1), (2), (3), (4), (5), (6), (7), (8)
 - przygotowywanie
 - przetwarzanie wielu nieznanymi faktur
 - rozwiązywanie problemów związanych z nieznanymi dostawcami
 - przeglądanie wyników instalacji domyślnej
 - wymagania wstępne
 - uruchomienie aplikacji
 - uruchomienie aplikacji na Datacap Web Client
 - uruchomienie czynności Batch Profiler (1), (2)
 - uruchomienie czynności Eksportu (1), (2)
 - uruchomienie czynności FlexID
 - uruchomienie czynności skanowania
 - na Datacap Web Client
 - uruchomienie czynności skanowania używając Datacap Desktop
 - uruchomienie czynności przesyłania
 - na Datacap Web Client
 - uruchomienie czynności Weryfikacji
 - na Datacap Desktop
 - na Datacap Web Client

- przykładowe obrazy (1), (2)
 - skanowanie obrazów faktur
 - na Datacap Web Client
 - skanowanie obrazów faktur używając Datacap Desktop
 - konfigurowanie dla przetwarzania faktur
 - rozpoczęcie pracy z Datacap Web Client
 - skróty czynności (1), (2)
 - przesyłanie obrazów faktur
 - na Datacap Web Client
 - Weryfikowanie instrukcji w oknie
 - weryfikowanie danych na fakturze
 - na Datacap Desktop
 - na Datacap Web Client
 - przeglądanie domyślnych uprawnień do stacji na kliencie Datacap Web Client
 - przeglądanie domyślnych uprawnień użytkownika na kliencie Datacap Web Client
 - przeglądanie domyślnych grup uprawnień użytkownika na kliencie Datacap Web Client
- APT_Localization, działanie
 - opis
- architektura
 - Datacap, aplikacje
- archiwizować
 - zadania wsadowe
- aspx, strony WWW
 - aVerify
 - ImgEnter
 - Restruct
 - VeriFine
- składanie dokumentów
- Przypisywanie nazwy pliku strony
 - PUT, metoda
 - SetPageFileName
- AssignFieldDefault, działanie (1), (2)
- Przypisywanie grupy do zadania wsadowego
 - Definiowanie niestandardowych filtrów monitora zadań
 - Dodawanie reguły w Datacap Studio
 - Dodawanie wartości niestandardowych w Application Manager
- Uwierzytelnianie
- uwierzytelnianie
 - konfigurowanie Rulerunner
 - dodawanie grup
 - dodawanie stacji
 - Datacap, uwierzytelnianie
 - nadawanie nazwy domeny
 - nadawanie nazwy grupy uprawnień
 - logowanie do Datacap Web Client
 - Datacap
 - rozwiązywanie problemów
- Uwierzytelnianie
 - silnik treści
 - Datacap Web Services
 - Użytkownik skonfigurowany
 - Użytkownik końcowy
 - Email, działania

- [Faks, działania](#)
 - [FileNetIDM](#)
 - [IBM Content Manager](#)
 - [SharePoint](#)
- System uwierzytelniania
 - [Planowanie w systemie Datacap](#)
- rejestracja automatyczna
 - [dopasowywanie wzorca](#)
 - [korzystanie z działania FindFingerprint](#)
- AutoDelete, proces
 - [usuwanie zadań wsadowych](#)
- AutoDelete, zestaw reguł
 - [usuwanie zadań wsadowych](#)
- AutoDelete, program narzędziowy
 - [usuwanie zadań wsadowych](#)
- Autodoc, działania
 - [BlankPagesIDBySize \(1\), \(2\)](#)
 - [CalculateOffset \(1\), \(2\)](#)
 - [CreateFingerprint \(1\), \(2\)](#)
 - [DeleteFingerprint \(1\), \(2\)](#)
 - [FindBlackFingerprint \(1\), \(2\)](#)
 - [FindFingerprint \(1\), \(2\)](#)
 - [FindTemplate \(1\), \(2\)](#)
 - [MergeCCOs_ByType \(1\), \(2\)](#)
 - [SetApplicationID \(1\), \(2\)](#)
 - [SetFilter_HostName \(1\), \(2\)](#)
 - [SetFilter_PageType \(1\), \(2\)](#)
 - [SetFingerprint \(1\), \(2\), \(3\)](#)
 - [SetFingerprint WebServiceURL](#)
 - [SetFingerprintDir](#)
 - [SetFingerprintFailureThreshold \(1\), \(2\)](#)
 - [SetFingerprintSearchArea \(1\), \(2\)](#)
 - [SetFingerprintWebServiceURL](#)
 - [SetMaxOffset \(1\), \(2\)](#)
 - [SetProblemValue \(1\), \(2\)](#)
 - [SetSearchArea \(1\), \(2\)](#)
 - [SetTemplateDir \(1\), \(2\)](#)
 - [UpdateFingerprintStats](#)
 - [UpdateFPStats](#)
- [Autofield](#)
- [AutoField, działanie](#)
- AutoFingerprint, zestaw reguł
 - [dodawanie do profilu czynności Weryfikacji \(1\), \(2\)](#)
 - [przypisywanie reguły do każdego typu stron](#)
 - [tworzenie](#)
 - [aktualizowanie](#)
- [automatyczne czynności wykonywane w tle](#)
- automatyczne przetwarzanie w tle
 - [aktualizowanie aplikacji TravelDocs](#)
 - [analizowanie dziennika Rulerunner](#)
 - [definiowanie czynności wykonywanych w tle](#)
 - [wyłączanie dziennika Rulerunner](#)
 - [włączanie rejestrowania Rulerunner](#)

- uruchamianie zadania wsadowego w przepływie pracy
 - konfigurowanie czynności wykonywanych w tle
 - konfigurowanie monitora zadań
- automatyczne generowanie odcisku
 - aktualizowanie aplikacji TravelDocs
 - dodawanie zestawu reguł do profilu czynności Weryfikowania (1), (2)
 - przypisywanie reguły do każdego typu stron
 - tworzenie zestawu reguł AutoFingerprint
 - przeglądanie pliku dziennika RRS
 - uruchamianie zadania wsadowego w przepływie pracy
- automatyczne aktualizacje poprawek
- AVerify, klient WWW
 - tworzenie paneli statycznych

B

- Czynności wykonywane w tle
 - [Medical Claims](#)
- czynności wykonywane w tle
 - [automatyzacja](#)
 - konfigurowanie Rulerunner
 - [konfigurowanie zabezpieczeń w folderze aplikacji](#)
 - [konfigurowanie zabezpieczeń w folderze Datacap](#)
 - [konfigurowanie zabezpieczeń w folderze RRS](#)
 - [konfigurowanie współużytkowania w folderze Datacap](#)
 - [definiowanie](#)
 - [przetwarzanie w Rulerunner](#)
 - [konfigurowanie](#)
 - [korzystanie z Rulerunner w celu automatyzacji](#)
- [BackupFolder](#)
- [BackupFolder, działanie](#)
- [klasyfikacja kodów kreskowych](#)
- Barcode_P, działania
 - [Get2DCodesBP \(1\), \(2\)](#)
 - [GetAllBarcodesBP \(1\), \(2\)](#)
 - [GetBarcodeBP \(1\), \(2\)](#)
 - [GetDataMatrixCodeBP \(1\), \(2\)](#)
 - [IdentifyByBarcodesBP](#)
 - [MatchBarcodeBP \(1\), \(2\)](#)
 - [MatchBarcodePrefixBP \(1\), \(2\)](#)
 - [ReadBarcodeBP](#)
 - [ReadBarCodeBP](#)
 - [SetMinimumConfidenceBP](#)
- Barcode_X, działania
 - [GetBarCode \(1\), \(2\)](#)
 - [MatchBarCode \(1\), \(2\)](#)
 - [ReadBarCode \(1\), \(2\)](#)
- Base64, kodowanie
 - POST, metoda
 - [SetFile](#)
 - [UploadFile](#)
- Atrybuty zadań wsadowych
 - GET, metoda
 - [GetBatchAttributes](#)

- Historia zadań wsadowych
 - GET, metoda
 - [GetBatchHistory](#)
- Identyfikator zadania wsadowego
 - GET, metoda
 - [GetBatchId](#)
- zarządzanie zadaniami wsadowymi
 - FastDoc [\(1\)](#), [\(2\)](#), [\(3\)](#)
- Batch Pilot, panele
 - przekształcanie do Datacap Desktop [\(1\)](#), [\(2\)](#), [\(3\)](#), [\(4\)](#), [\(5\)](#), [\(6\)](#), [\(7\)](#)
 - generowanie układu, pliki XML [\(1\)](#), [\(2\)](#)
 - przeglądanie układu, pliki XML [\(1\)](#), [\(2\)](#)
- przetwarzanie wsadowe
 - [aplikacja IIS, przetwarzanie puli](#)
- Przetwarzanie wsadowe, działania
 - [ProcessChange BatchStatus](#)
 - [ProcessChangeBatch StatusOrder](#)
 - [ProcessChangeBatch StatusTaskOrder](#)
 - [ProcessChangeBatchStatus](#)
 - [ProcessChangeBatchStatus TaskOrder](#)
 - [ProcessClearAuditTable](#) [\(1\)](#), [\(2\)](#)
 - [ProcessClearDebugTable](#) [\(1\)](#), [\(2\)](#)
 - [ProcessDeleteBatches](#) [\(1\)](#), [\(2\)](#)
 - [ProcessDeleteBatchesEx](#) [\(1\)](#), [\(2\)](#)
 - [ProcessInjectBatches](#) [\(1\)](#), [\(2\)](#)
 - [ProcessMoveBatches](#) [\(1\)](#), [\(2\)](#)
 - [ProcessMoveBatchesEx](#)
 - [ProcessMoveBatchesEX](#)
 - [ProcessMoveDBRecords](#) [\(1\)](#), [\(2\)](#)
 - [ProcessReset PendingOrNotify](#)
 - [ProcessResetPendingOrNotify](#)
 - [ProcessRunSqlQuery](#) [\(1\)](#), [\(2\)](#)
 - [ProcessRunSqlQueryEx](#)
- profil zadania wsadowego, konfiguracja
 - FastDoc
 - [Tryb lokalny](#)
- Batch Profiler, czynność
 - [TravelDocs](#)
- Zadanie wsadowe, tworzenie kolejek użytkowników i stacji
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
- zadanie wsadowe, status
 - monitorowanie [\(1\)](#), [\(2\)](#)
 - raportowanie w [\(1\)](#), [\(2\)](#)
- zadania wsadowe
 - [archiwizowanie](#)
 - [zmiana kolejności](#)
 - [zmiana statusu](#)
 - [sprawdzanie integralności strukturalnej](#)
 - tworzenie w FastDoc
 - [za pomocą skanera](#)
 - [przy użyciu zeskanowanych obrazów](#)
 - [usuwanie](#)

- FastDoc
- generowanie za pomocą VScan
- indeksowanie w FastDoc
- spełnianie poszczególnych kryteriów
- monitorowanie (1), (2)
- otwieranie w FastDoc
- przygotowanie do weryfikacji
- przetwarzanie w Datacap Studio
- przeglądanie w Datacap Desktop
- wykonywania pracy w przepływie pracy (1), (2), (3), (4), (5), (6)
- obsługa problemów z integralnością dokumentów
- folder wykonawczy
- status (1), (2)
- zatwierdzanie
- testowanie czynności eksportu
- przesyłanie (1), (2)
- przesyłanie w trybie offline
- weryfikowanie w FastDoc
- BatchVariable_ExportValue, działanie (1), (2)
- zadania wsadowe
 - błędy sprawdzania poprawności
- BlankFields, działanie (1), (2)
- BlankLines, działanie (1), (2)
- BlankPagesIDBySize, działanie (1), (2), (3)
- Box Connector
 - przesyłanie plików, przykłady
- Box Connector, działania
 - ustawienia parametrów
- BPilot, działanie (1), (2)
- odgałęzienie
 - konfigurowanie (1), (2)
 - definiowanie warunków
 - opis
 - zgłaszanie flag warunków
- punkty zatrzymania
 - czyszczenie
 - wyłączenie
 - pełne
 - ogólne
 - ustawienie
 - typy
- potrzeby biznesowe
 - projektowanie
 - określanie wartości pól
- sprawdzanie wiarygodności biznesowej, reguły

C

- C2BW_Convert, działanie (1), (2)
- C2BW_SetAttributes, działanie (1), (2)
- Obliczanie, działanie (1), (2)
- obliczane pola
 - sprawdzanie poprawności
- CalculateDateDifference, działanie (1), (2)

- CalculateFields, działanie (1), (2)
- CalculateHCFALineCharges, działanie
- CalculateInvoiceTotalLocalized, działanie
 - opis
- CalculateLineItemLocalized, działanie
 - opis
- CalculateLocalOffset, działanie (1), (2)
- CalculateNotesZone, działanie (1), (2)
- CalculateOffset, działanie (1), (2)
- CalculateUBLLineCharges, działanie
- obliczanie kosztu, pola
 - tworzenie reguł sprawdzania wiarygodności
- CallPOLR, działanie (1), (2)
- CaptureOpInfo, działanie (1), (2)
- Typ samochodu, pole
 - zapobieganie nadpisywaniu
- Typ_samochodu, pole
 - attaching a dictionary
- typ samochodu, reguła
 - tworzenie reguł sprawdzania wiarygodności
- Case Manager
 - integrowanie
- CC, działania
 - ClassifyCC (1), (2)
 - ClassifyTextCC (1), (2)
 - FindFingerprintCC
 - RunDecisionPlanCC (1), (2)
 - RunDecisionPlanForBlocksCC
 - RunDecisionPlanForTextCC (1), (2)
 - SetDecisionPlanCC (1), (2)
 - SetDecisionPlanFieldsCC (1), (2)
 - SetKnowledgeBaseCC (1), (2)
 - SetLanguageCC (1), (2)
 - SetListenerURLCC (1), (2)
 - SetProblemValueCC (1), (2)
 - UpdateKnowledgeBaseCC (1), (2)
- Cco2cco, działania
 - Cco2cco (1), (2), (3), (4)
 - NormalizeCCO (1), (2)
 - SetMaxCharacter HeightAVG
 - SetMaxCharacter HeightTMM
 - SetMaxCharacterHeightAVG
 - SetMaxCharacterHeightTMM
- CCNormalization_OFF, działanie (1), (2)
- Zmiana hasła
 - POST, metoda
 - ChangeUserPassword
- Zmiana odwołań do aplikacji
 - Plik Datacap.xml (1), (2)
- Zmiana ustawień SSL
 - Datacap Web Client
 - server.ini
- pozycja znaku

- DCO, metody (1), (2)
 - [GetPosition](#)
 - [SetPosition](#)
- wartości znaków
 - [DCO, metody](#)
 - [set_CharValue](#)
- CharConfidence
 - [DCO, właściwości](#)
- wykresy
 - [dodawanie do raportu](#)
- CharValue
 - [DCO, właściwości](#)
- Sprawdzenie integralności zadań wsadowych
 - GET, metoda
 - [CheckIntegrity](#)
- pole wyboru, opcje
 - [tworzenie słowników](#)
- pole wyboru, wiadomość
 - [ustanowienie pól nadrzędnych](#)
 - [metody](#)
 - [OCR/A](#)
 - [metoda oceny progu pikseli](#)
 - [ustawianie zmiennych pól nadrzędnych](#)
- sprawdzanie, typ zaznaczenia
 - [określanie](#)
- sprawdzanie, przetwarzanie
 - [IBM Datacap Advanced Handwriting Recognition](#)
- sprawdzanie, działania przetwarzania
 - [CheckProcessingCanada](#) (1), (2)
 - [CheckProcessingFrance](#)
 - [CheckProcessingFrance, działanie](#)
 - [CheckProcessingIndia](#) (1), (2)
 - [CheckProcessingUK](#) (1), (2)
 - [CheckProcessingUS](#) (1), (2)
- CheckAllIntegrity
 - [korzystanie](#)
- CheckAllIntegrity action (1), (2), (3)
- CheckAndFixDecimal, działanie (1), (2)
- CheckAndFixLocalDecimal, działanie
 - [opis](#)
- CheckDocCount, działanie (1), (2)
- [CheckDocID, działanie](#)
- CheckForSticky, działanie (1), (2)
- CheckFreeDiskSpace, działanie (1), (2), (3)
- [CheckFreeSpace, działanie](#)
- [sprawdzanie poprawności formatu](#)
- CheckIntegrity
 - [DCO, metody](#)
- Lista kontrolna
 - Dodawanie listy aplikacji
 - [Datacap.xml, plik](#)
 - Dodawanie listy aplikacji do pliku Datacap.xml
 - [Datacap, serwer](#)

- Dodawanie lokalizacji pliku Datacap.xml
 - [Datacap Web Client](#)
- Dodawanie zabezpieczeń
 - [folder aplikacji Datacap](#)
 - [Datacap Web Client \(1\), \(2\)](#)
- Dodawanie uprawnień do folderu aplikacji
 - [Datacap, serwer](#)
- Konfigurowanie
 - [Datacap, stacja robocza programisty](#)
 - [Fingerprint, usługa](#)
 - [aplikacja Rulerunner, czynności](#)
- Konfiguracja i testowanie IE
 - [Datacap Web Client \(1\), \(2\)](#)
- Konfiguracja i testowanie połączenia zdalnego
 - [Utworzenie pakietu](#)
 - [Internet Explorer](#)
- Konfigurowanie uwierzytelniania
 - [Rulerunner](#)
- [Konfigurowanie uprawnień do konta Rulerunner](#)
- Konfigurowanie
 - [Datacap, usługa serwisowa](#)
- Kopiowanie aplikacji
 - [Datacap, serwer](#)
- [Datacap Web Server, konfiguracja](#)
- Instalowanie
 - [Datacap, serwer](#)
- Instalowanie i konfigurowanie
 - [Datacap, serwer](#)
- Instalacja i konfiguracja
 - [Rulerunner](#)
- Zdalna stacja robocza, konfiguracja
 - [Datacap Web Client](#)
- Uruchamianie klienta, aplikacja
 - [Datacap Web Client](#)
- Testowanie instalacji
 - [Fingerprint, usługa](#)
- [Stacja robocza użytkownika, konfiguracja](#)
- Web Client Configuration, narzędzie
 - [Datacap Web Client](#)
 - [Opakowanie](#)
- [CheckPageCount, działanie \(1\), \(2\)](#)
- [CheckProcessingCanada, działanie \(1\), \(2\)](#)
- [CheckProcessingFrance, działanie \(1\), \(2\)](#)
- [CheckProcessingIndia, działanie \(1\), \(2\)](#)
- [CheckProcessingUK, działanie \(1\), \(2\)](#)
- [CheckProcessingUS, działanie \(1\), \(2\)](#)
- [CheckSubFields, działanie \(1\), \(2\)](#)
- Chiński
 - [kody języków](#)
- [ChkConfidence, działanie \(1\), \(2\)](#)
- [ChkDCOStatus, działanie](#)
- [ChkDCOStatus, działanie](#)
- [ChkDCOType, działanie \(1\), \(2\)](#)

- ChkIntegrity, działanie (1), (2)
- ChkLastDCOType, działanie (1), (2)
- Klasyfikacja
 - [Datacap Navigator](#)
- ClassifyCC, działanie (1), (2)
- ClassifyLayout, działania
 - [Opinia](#) (1), (2)
 - [Identyfikacja](#) (1), (2)
- ClassifyTextCC, działanie (1), (2)
- czyszczenie obrazów (1), (2)
- Czyszczenie
 - [DCO, metody](#)
- ClearAltText, działanie (1), (2)
- ClearCurrentField, działanie
 - [opis](#)
- ClearDCO, działanie (1), (2)
- ClearErrorMsg, działanie (1), (2), (3)
- [czyszczenie punktów zatrzymania](#)
- [czyszczenie ostatniego błędu](#)
- ClearLookupResults, działanie (1), (2)
- [Klient/serwer, konfiguracja](#)
- klient serwer, instalacja
 - [konfigurowanie Report Viewer](#)
- CloseConnection, działanie (1), (2)
- CloseDatabase, działanie (1), (2)
- CloseExportFile, działanie (1), (2)
- CMISClient, działania
 - [CMISCreateFolder](#) (1), (2)
 - [CMISDeleteFile](#) (1), (2)
 - [CMISDeleteFolder](#) (1), (2)
 - [CMISDoesFileExist](#) (1), (2)
 - [CMISDoesFolderExist](#) (1), (2)
 - [CMISDownloadFile](#) (1), (2)
 - [CMISLogDocumentTypes](#) (1), (2)
 - [CMISLogin](#) (1), (2)
 - [CMISRefreshClientCache](#) (1), (2)
 - [CMISSetDocUploadProperty](#) (1), (2)
 - [CMISSetDocUploadType](#) (1), (2)
 - [CMISSetVersion](#)
 - [CMISUploadFile](#) (1), (2)
 - [CMISUploadPage](#) (1), (2)
 - [opis](#)
- [CMISCreateFolder](#), działanie (1), (2)
- [CMISDeleteFile](#), działanie (1), (2)
- [CMISDeleteFolder](#), działanie (1), (2)
- [CMISDoesFileExist](#), działanie (1), (2)
- [CMISDoesFolderExist](#), działanie (1), (2)
- [CMISDownloadFile](#), działanie (1), (2)
- [CMISLogDocumentTypes](#), działanie (1), (2)
- [CMISLogin](#), działanie (1), (2)
- [CMISRefreshClientCache](#), działanie (1), (2)
- [CMISSetDocUploadProperty](#), działanie (1), (2)
- [CMISSetDocUploadType](#), działanie (1), (2)

- [CMISSetVersion](#), działanie
- [CMISUploadFile](#), działanie (1), (2)
- [CMISUploadPage](#), działanie (1), (2)
- kod
 - [przejsięcie w jednym kroku](#)
- [ColorToBW](#), działania
 - [C2BW_Convert](#) (1), (2)
 - [C2BW_SetAttributes](#) (1), (2)
- kolumna, stopki
 - [dodawanie agregacji](#)
- kolumna, nazwa
 - [zmiana](#)
- [CombinePreviousDoc](#), działanie
 - [opis](#)
- wiersz komendy, opcje
 - [klonowanie, aplikacje](#)
 - [migrowanie, aplikacje](#)
 - [aktualizowanie, aplikacje](#)
- wiersz komendy, parametry
 - [Datacap, instalacja](#)
- Najczęściej wykonywane działania
 - [ExceptionSetFileTypes](#)
 - [ExceptionSetHandler](#)
 - [ExceptionSetTaskCondition](#)
 - [ExceptionSetVariableName](#)
 - [SetNamePattern](#)
- często wykorzystywane parametry
 - [przełądanie](#)
- [CommonParseAddress](#), działanie
- [CommonValAddress](#), działanie
- [CompareFields](#), działanie (1), (2)
- [CompareFieldsText](#), działanie
- warunek, flagi
 - [zgłaszanie dla odgałęzień](#)
 - [zgłaszanie dla dzielenia](#)
- warunki
 - [definiowanie dla odgałęzień](#)
 - [definiowanie dla dzielenia](#)
- [Ufność](#)
- ufność, poziom
 - [ustawienie dopasowania wzorca](#)
- ufność, poziomy
 - [opis](#)
 - [nadpisywanie domyślnych wartości](#)
 - [strony wykonawcza](#)
- [ConfidenceString](#)
 - [DCO, właściwości](#)
- konfiguracja
 - [Datacap](#)
 - [Datacap Web Client, przysyłanie, usługa](#) (1), (2)
 - [DB2, bazy danych](#)
 - [planowanie](#)
 - [Email Connector, działania](#)

- Fax Connector, działania
- FileNet Image Services Connector, działania
- FileNet P8 Connector (1), (2)
- Fingerprint Service
- zbieranie informacji konfiguracyjnych o Rulerunner
- IBM Content Manager Connector, działania
- Microsoft SQL Server, bazy danych
 - planowanie
- mobilna
- monitorowanie zadań wsadowych w Rulerunner (1), (2)
- Bazy danych Oracle
 - planowanie
- wymagania wstępne
- Report Viewer (1), (2)
 - przegląd (1), (2)
 - wymagane czynności (1), (2)
- Rulerunner (1), (2), (3), (4)
 - Datacap, uwierzytelnianie
 - Datacap Server, uprawnienia
 - wymagania wstępne
 - zamykanie oprogramowania Datacap
- Rulerunner, uwierzytelnianie
 - dodawanie grup
 - dodawanie stacji
 - nadawanie nazwy domeny
 - nadawanie nazwy grupy uprawnień
 - logowanie do Datacap Web Client
- uruchamianie aplikacji za pomocą Rulerunner
- uruchamianie czynności za pomocą Rulerunner
- SharePoint Connector
- konfigurowanie baz danych
 - DB2
 - Oracle, bazy danych
 - SQL Server, bazy danych
- plik konfiguracyjny
 - edytowanie w Datacap Web Client, przesyłanie, usługa
- ustawienia konfiguracyjne
 - Datacap Maintenance Manager
- konfigurowanie
 - Datacap Maintenance Manager
 - Fingerprint Maintenance Tool
 - Rulerunner
 - na jednym komputerze
- Konfigurowanie
 - Lista kontrolna
 - Datacap, stacja robocza programisty
 - Rulerunner
 - Datacap Web Server
 - Windows 2008 IIS 7.5 Server
 - Windows Server 2012 R2
- Konfiguracja i testowanie IE
 - Lista kontrolna
 - Datacap Web Client (1), (2)

- Konfiguracja i testowanie połączenia zdalnego
 - [Lista kontrolna](#)
 - [Internet Explorer](#)
- konfigurowanie Internet Explorer
 - [Datacap Web Client](#)
 - [na jednym komputerze, instalacja](#)
- Konfigurowanie Internet Explorer
 - [Programista, stacja robocza](#)
- [konfigurowanie zdalnego klienta skanowania](#)
- [konfigurowanie zdalnego klienta skanowania](#)
- [konfigurowanie](#)
 - [AIndex, klient WWW](#)
 - [struktura zadania wsadowego w Datacap Navigator](#)
 - [Datacap Navigator \(1\), \(2\), \(3\), \(4\), \(5\), \(6\)](#)
 - [DCO Tree w Datacap Navigator](#)
 - [Medical Claims](#)
 - [panele w Datacap Navigator \(1\), \(2\)](#)
 - [wymagania wstępne](#)
 - [ProtoId, klient WWW](#)
 - [Rulerunner](#)
 - [Verifine, klient WWW](#)
- Konfigurowanie
 - [Lista kontrolna](#)
 - [Datacap, usługa serwisowa](#)
 - [Fingerprint, usługa](#)
- Konfigurowanie i testowanie
 - [Datacap Web Client](#)
 - [Zdalna stacja robocza](#)
- [Konfigurowanie opcji pulpitu](#)
- Konfigurowanie zewnętrznego authenticationLLLDAP
 - [ADLDS](#)
 - [ADSI](#)
 - [Datacap Server Manager](#)
 - [Datacap Server, usługa](#)
 - [LDAP](#)
- konfigurowanie Internet Explorer
 - [na jednym komputerze](#)
- Konfigurowanie Internet Explorer
 - [Datacap Web Client](#)
- Konfigurowanie zabezpieczeń
 - [Datacap Web Client](#)
 - [zaufana strona Opcje internetowe](#)
- konfigurowanie Datacap Web Client, serwer (1), (2)
- [konfigurowanie bazy danych do eksportu](#)
- [Konfigurowanie czynności przesyłania](#)
- Konfigurowanie serwisu Web Client Configuration
 - [Datacap Web Client](#)
- Połączenia, działanie (1), (2)
- [podłączenie do aplikacji](#)
- łańcuchy połączeń
 - [ustawienia zaawansowane](#)
 - [baza danych](#)
 - [odwołanie z działań](#)

- zapisywanie w pliku .app
- [ConnectOnConnectionError](#)
- [ConnectOnFaxImportError](#)
- Connector, działania
 - instalacja
 - weryfikowanie
 - pliki dziennika
 - przegląd
- Connector, działania
 - przegląd
- konektory
 - zawieranie w aplikacjach
- Silnik treści
 - [Uwierzytelnianie](#)
- [ContinueOn FaxImportError](#), działanie
- [ContinueOnConnectionError](#), działanie
- [ContinueOnFaxImportError](#), działania
 - [Connect](#)
- przekształcanie
 - Application Wizard (1), (2)
 - aplikacje (1), (2), (3), (4), (5), (6), (7)
 - tworzenie paneli Datacap Desktop (1), (2)
 - układ, pliki XML (1), (2)
- Przekształcanie, działania
 - Najczęściej wykonywane
 - [ExceptionSetFileTypes](#)
 - [ExceptionSetHandler](#)
 - [ExceptionSetTaskCondition](#)
 - [ExceptionSetVariableName](#)
 - [SetNamePattern](#)
 - Excel
 - [ExcelAutoFitColumns](#)
 - [ExcelAutoFitRows](#)
 - [ExcelOrientationToLandscape](#)
 - [ExcelOrientationToPortrait](#)
 - [ExcelPrintBlankPage](#)
 - [ExcelPrintGridlines](#)
 - [ExcelPrintQuality](#)
 - [ExcelScalingFactor](#)
 - [ExcelTiffCompression](#)
 - [ExcelWorkbookToImage](#)
 - Html
 - [HtmlPrintQuality](#)
 - [HtmlTiffCompression](#)
 - [HtmlToImage](#)
 - Obrazy
 - [ImageDefaultDPI](#)
 - [ImageFileTypesToConvert](#)
 - [ImageMonoThreshold](#)
 - [ImageMonoType](#)
 - [ImageToTIFF](#)
 - Outlook
 - [OutlookMessageTo AttachmentOnly](#)

- OutlookMessageToImage AndAttachment
 - OutlookPrintQuality
 - OutlookTiffCompression
 - Pdf
 - PDFBitDepth
 - PDFCompression
 - PDFConversionMethod
 - PDFDocumentToImage
 - PDFGrayscale
 - PDFHorizontalResolution
 - PDFQuality
 - PDFVerticalResolution
 - PdfFRE
 - PDFConversion Mode
 - PDFDocumentTo Image
 - PDFImage Compression
 - PDFImageFile Extension
 - PDFImageFile R esolution
 - PDFImageUse FastBinarization
 - PDFJpegQuality
 - Rtf
 - RtfPrintQuality
 - RtfTiffCompression
 - RtfToImage
 - Tiff
 - SplitMultipageTiff
 - SplitTIFFCompression
 - Txt
 - TxtPrintQuality
 - TxtTiffCompression
 - TxtToImage
 - Word
 - WordDocumentToImage
 - WordPrintQuality
 - WordTiffCompression
 - Zip
 - ZipOverwrite
 - ZipPassword
 - ZipUnpack
- ConvertEuroDateToUS, działanie
 - opis
- ConvertFieldTo Currency, działanie
- ConvertFieldToCurrency, działanie
- ConvertGraytoBW, działanie (1), (2)
- ConvertHyphen, działanie
- ConvertToJPEG, działanie (1), (2)
- ConvertToLowerCase, działanie (1), (2)
- ConvertToTIFF, działanie (1), (2)
- ConvertToUpperCase, działanie (1), (2)
- ConvertUSDateToEuro, działanie
 - opis
- Kopiowanie aplikacji
 - Lista kontrolna

- [Datacap, serwer](#)
- Kopiowanie plików do folderu pamięci podręcznej
 - POST, metoda
 - [CopyFilesToCache](#)
- [CopyAllBlocks](#), działanie
- [CopyDirectory](#), działanie (1), (2)
- [CopyField](#), działanie (1), (2)
- [CopyFieldToField](#), działanie (1), (2)
- [CopyFile](#), działanie (1), (2), (3), (4)
- [kopiowanie aplikacji na serwer Datacap](#)
- [kopiowanie aplikacji na serwer Datacap Server](#)
- [Kopiowanie aplikacji](#)
- [CopyLabelValuePairs](#), działanie
- [CopyPD2DD](#), działanie (1), (2)
- [CountPagesToDocumentVar](#), działanie
- [CountPagesToDocVar](#), działanie
 - [opis](#)
- Tworzenie rekordu zadania wsadowego
 - POST, metoda
 - [CreateBatch](#)
- tworzenie konta
 - [Datacap Maintenance Manager](#)
- tworzenie bazy danych
 - [DB2](#)
 - [DB2, użytkownicy](#)
 - [Microsoft SQL Server, użytkownicy](#)
 - [Oracle](#)
 - [Oracle, użytkownicy](#)
 - [SQL Server](#)
- [tworzenie typu dokumentu](#)
- [tworzenie zestawu danych ExportDB](#)
- [tworzenie zestawu danych ExportXML](#)
- [tworzenie klas odcisków](#)
- [tworzenie reguły ManualIDValidate](#)
- Tworzenie lub kopiowanie aplikacji
 - [aplikacja oparta na usłudze CMIS](#)
- Tworzenie lub upewnianie się, że domena/konto w systemie Windows istnieje
 - [Datacap Web Client](#)
- Tworzenie lub zapisywanie czynności
 - POST, metoda
 - [SaveTask](#)
- [tworzenie odcisków na stronie](#)
- [tworzenie czynności](#)
- [tworzenie słownika](#)
- [CreateBatchSubfolder](#) (1), (2)
- [CreateBatchSubfolder](#), działanie
- [CreateBlockCCO](#), działanie (1), (2)
- [CreateCcoFromLayout](#), działanie
- [CreateDocs](#), czynność
 - [konfigurowanie Rulerunner przed uruchomieniem](#)
 - [tworzenie](#)
- [CreateDocuments](#)
 - [DCO](#), metody (1), (2)

- CreateDocuments, działanie (1), (2)
- CreateFields, działanie (1), (2)
- CreateFingerprint, działanie (1), (2), (3), (4)
- CreateFolder, działanie (1), (2)
- CreateHTML, działanie
- CreateNew, działanie
- CreateTextFile, działanie (1), (2)
- tworzenie
 - Datacap Web Client, serwis (1), (2)
- Tworzenie kolumny niestandardowej
 - Monitor zadań
- tworzenie czynności zdalnego skanowania
- tworzenie pól danych
- tworzenie obiektów w polach
 - CreateFields
 - DCO, metody
- tworzenie plików odcisków
- tworzenie plików danych na stronie
- tworzenie typów stron
- tworzenie paneli statycznych
 - AVerify, klient WWW
- Tworzenie czynności skanowania
 - Datacap Web Client
- tworzenie stref
 - inne typy stron
 - TravelDocs
- pola waluty
 - tworzenie reguł sprawdzania wiarygodności
 - sprawdzanie poprawności
- strony niestandardowe
 - tworzenie i używanie
- panele niestandardowe
 - Datacap Navigator (1), (2)
 - określanie do wykorzystania w czynności
- niestandardowe szablony raportu
 - tworzenie dla Report Viewer (1), (2)
- raporty niestandardowe
 - przeglądanie w Przeglądarce raportów
- dostosowywanie do Datacap Navigator
- dostosowywanie układu panelu

D

- statystyki dzienne
 - przeglądanie na FastDoc
- dane
 - zmienianie nazw pól
 - obliczanie średniej
 - eksportowanie do bazy danych
 - eksportowanie do pliku tekstowego
 - eksportowanie do pliku XML
 - filtrowanie
 - filtrowanie raportu (1), (2)
 - wykrywanie na niezidentyfikowanej stronie

- konfigurowanie filtrów
 - sprawdzanie poprawności
- pola danych
 - tworzenie (1), (2)
- format danych
 - określanie do eksportu
- dane, lokalizowanie
 - dopasowywanie tekstu
- rozpoznawanie danych
- dane, sprawdzanie poprawności
 - obliczane pola
 - definicja
 - wyświetlanie niepowodzeń operatorowi
 - przykłady
 - korzystanie ze źródeł zewnętrznych
- weryfikacja danych
 - Datacap Desktop
 - Datacap Web Client
 - podwójnie ślepa próba
 - pola
- baza danych
 - ustawienia zaawansowane
 - łańcuchy połączeń
- baza danych, połączenia
 - testowanie
 - weryfikowanie
- baza danych, dostawcy
 - zmiany w Datacap Application Copy Tool
- baza danych, uprawnienia dostępu
 - przypisywanie
- baza danych, struktury
 - tworzenie pojedynczego użytkownika DB2
 - tworzenie pojedynczego użytkownika serwera Microsoft SQL Server
 - tworzenie pojedynczego użytkownika Oracle
 - definiowanie
- bazy danych
 - zmiana dostawców baz danych
 - opcje wiersza komend
 - konfigurowanie
 - DB2 (1), (2)
 - Microsoft SQL Server
 - Oracle (1), (2)
 - SQL Server
 - tworzenie
 - tworzenie pojedynczego użytkownika DB2
 - tworzenie pojedynczego użytkownika serwera Microsoft SQL Server
 - tworzenie pojedynczego użytkownika Oracle
 - Datacap Application Copy Tool
 - migracja (1), (2)
 - migracja do nowego dostawcy
 - opcje migrowania
 - przenoszenie
 - przenoszenie do nowego środowiska (1), (2)

- planowanie
 - DB2
 - Microsoft SQL Server
 - Oracle
 - uprawnienia zabezpieczeń
 - aktualizacja w istniejących środowiskach
 - aktualizacja (1), (2), (3)
 - dostęp użytkownika
 - interfejs użytkownika
- Bazy danych
 - Planowanie w systemie Datacap
- databasesDatacap Application Copy Tool
 - łańcuchy połączeń
- Datacap
 - architektura
 - konfigurowanie
 - konfigurowanie uwierzytelniania systemu Windows
 - tworzenie hierarchii dokumentów
 - instalowanie i konfigurowanie
 - lokalizowanie, aplikacje
 - pliki dziennika
 - otwieranie aplikacji przykładowej
 - przegląd
 - planowanie instalacji
 - planowanie użytkowników i grup
 - usuwanie (1), (2)
 - ponowne uruchamianie
 - uruchamianie na stacji roboczej programisty
 - instalacja autonomiczna
 - zatrzymywanie
 - przepływy pracy
- Datacap, dostęp
 - DB2, bazy danych
 - Microsoft SQL Server, bazy danych
 - Oracle, bazy danych
- Datacap Accounts Payable
 - dodawanie elementów wiersza zamówienia
 - dodawanie dostawców do listy Demo Vendors
 - powiązanie dostawcy z odciskiem
 - przechwytywanie pól na obróconym obrazie
 - czyszczenie obrazów faktur (1), (2)
 - opis
 - eksportowanie obrazów faktur (1), (2)
 - nadawanie uprawnień
 - obsługa nieznanymi faktur
 - identyfikowanie wierszy szczegółów na nieznanymi fakturach
 - logowanie do Datacap Web Client
 - wstępnie skonfigurowane zadania (1), (2)
 - przygotowywanie obrazów faktur
 - przetwarzanie obrazów faktur (1), (2), (3), (4), (5), (6), (7), (8)
 - przygotowywanie
 - przetwarzanie wielu nieznanymi faktur
 - rozwiązywanie problemów związanych z nieznanymi dostawcami

- przeglądanie wyników instalacji domyślnej
 - wymagania wstępne
- uruchomienie aplikacji
- uruchomienie aplikacji na Datacap Web Client
- uruchomienie czynności Batch Profiler (1), (2)
- uruchomienie czynności Eksportu (1), (2)
- uruchomienie czynności FlexID
- uruchomienie czynności skanowania
 - na Datacap Web Client
- uruchomienie czynności skanowania używając Datacap Desktop
- uruchomienie czynności przesyłania
 - na Datacap Web Client
- uruchomienie czynności Weryfikacji
 - na Datacap Desktop
 - na Datacap Web Client
- przykładowe obrazy (1), (2)
- skanowanie obrazów faktur
 - na Datacap Web Client
- skanowanie obrazów faktur używając Datacap Desktop
- konfigurowanie dla przetwarzania faktur
- rozpoczynanie pracy z Datacap Web Client
- skróty czynności (1), (2)
- przesyłanie obrazów faktur
 - na Datacap Web Client
- Weryfikowanie instrukcji w oknie
- weryfikowanie danych na fakturze
 - na Datacap Desktop
 - na Datacap Web Client
- przeglądanie domyślnych uprawnień do stacji na kliencie Datacap Web Client
- przeglądanie domyślnych uprawnień użytkownika na kliencie Datacap Web Client
- przeglądanie domyślnych grup uprawnień użytkownika na kliencie Datacap Web Client
- Datacap Accounts Payable, działania
 - APT_Localization (1), (2)
 - APTCustom
 - AddAllTaxesToTaxField
 - AddToDate
 - CalculateInvoice TotalLocalized
 - CalculateLineItemLocalized
 - ClearCurrentField
 - ConvertEuroDateToUS
 - ConvertUSDateToEuro
 - FindTaxValue
 - IsDate_FormatEuro
 - IsInvoiceFromUS
 - MakeFieldHighConfidence
 - PopulateTaxType
 - CheckAndFixLocalDecimal
 - ConcatLineValues
 - MergeLineItem FieldToPageField
 - MergePageFieldToDocVar
 - Dokumenty
 - CombinePreviousDoc
 - CountPagesToDocVar

- [IsFirstDocInBatch](#)
 - [RemoveDocumentStructure](#)
 - [FlexID](#)
 - [RunFlexIDPanel](#)
 - [Intellocate_Learning](#)
 - [Learn_Zones](#)
 - [Learn_ZonesFPX](#)
 - [IsFieldLocalCurrency](#)
 - [IsLocalDecimalSeparator](#)
 - [IsOriginalEuroFormat](#)
 - [IsWorkstationLocale](#)
 - [PageID](#)
 - [PageIDByBCSep](#)
 - [PageIDBySeqTypes](#)
 - [PageIDByVariableChange](#)
 - [PreVerifySetup](#)
 - [SetLabels](#)
 - [Redagowanie](#)
 - [EraseRect](#)
 - [GetAllBarcodes](#)
 - [RedactByRegEx](#)
 - [RedactField](#)
- [Datacap Application Copy Tool](#)
 - [zmiana dostawców baz danych](#)
 - [interfejs wiersza komend](#)
 - [łańcuchy połączeń](#)
 - [migracja baz danych](#)
 - [przenoszenie aplikacji \(1\), \(2\)](#)
 - [przenoszenie baz danych](#)
 - [aktualizowanie aplikacji \(1\), \(2\)](#)
 - [aktualizowanie baz danych](#)
 - [interfejs użytkownika](#)
- [Datacap, folder aplikacji](#)
 - [Dodawanie zabezpieczeń](#)
 - [Lista kontrolna](#)
- [Datacap\aplikacja, folder](#)
 - [Właściwości Datacap](#)
 - [Ustawianie zabezpieczeń \(1\), \(2\)](#)
- [Datacap Application Manager](#)
 - [konfigurowanie profili czynności TravelDocs](#)
- [Aplikacje Datacap](#)
 - [architektura](#)
 - [tworzenie](#)
 - [w kreatorze Application Wizard](#)
 - [Tworzenie aplikacji](#)
 - [Datacap Studio](#)
 - [określanie formatu danych](#)
- [Datacap, zadania wsadowe](#)
 - [przesyłanie \(1\), \(2\)](#)
- [Datacap, klient](#)
 - [Instalowanie](#)
- [Datacap Client](#)
 - [instalowanie komponentów dla programistów \(1\), \(2\)](#)

- Datacap, komponenty
 - [Zastosowanie licencji](#)
 - [Instalowanie](#)
- Datacap, bazy danych
 - [dostęp użytkownika](#)
- Datacap Desktop (1), (2), (3), (4)
 - [Konfigurowanie aplikacji przed uruchomieniem](#)
 - [Włączenie rejestrowania dla rozwiązywania problemów](#)
 - [Wydruk dokumentu, skanowanie lokalnie](#)
 - Skróty klawiszowe
 - [Poprawka, czynność](#)
 - [Skan, czynność](#)
 - [Weryfikowanie, czynność](#)
 - [Otwieranie zadania wsadowego](#)
 - [Uruchamianie czynności](#)
 - [Uruchamianie czynności Batch Profiler](#)
 - [Uruchamianie czynności eksportu](#)
 - [Uruchamianie czynności weryfikowania](#)
 - [Uruchamianie czynności VScan](#)
 - [Uruchamianie czynności PageID](#)
 - [Uruchamianie czynności skanowania](#)
 - [skanowanie](#)
 - [Skanowanie stron z wydrukowanych dokumentów](#)
 - [Weryfikowanie stron](#)
- Datacap Desktop, panele
 - [przekształcanie z Batch Pilot \(1\), \(2\), \(3\), \(4\), \(5\), \(6\), \(7\)](#)
 - [przekształcanie z DotEdit \(1\), \(2\), \(3\)](#)
 - [tworzenie paneli Datacap Desktop \(1\), \(2\)](#)
 - [generowanie układu, pliki XML \(1\), \(2\)](#)
 - [przeglądanie układu, pliki XML \(1\), \(2\)](#)
- Datacap, stacja robocza programisty
 - [Konfigurowanie](#)
 - [Lista kontrolna](#)
- folder Datacap
 - [Właściwości Datacap](#)
 - [Konfigurowanie zabezpieczeń](#)
 - [Konfigurowanie uprawnień współużytkowania](#)
 - [Datacap Web Services](#)
 - [konfigurowanie zabezpieczeń](#)
 - [konfigurowanie współużytkowania](#)
 - [Uprawnienia współużytkowania](#)
- Datacap, instalowanie
 - [często wykorzystywane parametry](#)
 - [uruchamianie z wiersza komend](#)
- Datacap, instalacja
 - [parametry wiersza komend](#)
 - [Microsoft Windows Installer, parametry](#)
 - [uruchamianie z wiersza komend \(1\), \(2\)](#)
- Datacap Maintenance Manager
 - [Uprawnienia dostępu do konta](#)
 - [c:\Datacap, folder](#)
 - [Datacap\NENU, zabezpieczenia folderu](#)
 - [Datacap\RRS, folder](#)

- Monitorowany folder aplikacji
 - konto, ustawienia
 - konto, uprawnienia współużytkowania
 - dodawanie działań do
 - AutoDelete, proces
 - AutoDelete, zestaw reguł
 - AutoDelete, program narzędziowy
 - ustawienia konfiguracyjne
 - konfigurowanie
 - tworzenie kont
 - tworzenie kont dla
 - usuwanie zadań wsadowych (1), (2), (3)
 - usuwanie zestawów reguł z
 - instalowanie
 - otwieranie aplikacji
 - przegląd
 - ręczne uruchamianie zestawów reguł
 - ustawianie uprawnień udostępniania konta
 - konfiguracja
- Datacap Maintenance Manager, działania
 - przegląd
- Datacap Maintenance Manager, aplikacje
 - tworzenie
 - Datacap Studio
 - Datacap Maintenance Manager
 - ręczne uruchamianie zestawów reguł
 - Windows Task Scheduler
 - automatyczne uruchamianie zestawów reguł
- Datacap Navigator
 - dodawanie czynności z Datacap Navigator
 - administracja (1), (2)
 - Administracja
 - Klasyfikowanie
 - Konfigurowanie grup i użytkowników
 - Konfigurowanie skrótów
 - konfigurowanie (1), (2)
 - konfigurowanie adresu URL dla
 - dostosowywanie monitora zadań
 - dostosowywanie układu strony
 - domyślny układ strony
 - dane zewnętrzne, usługi (1), (2)
 - instalowanie (1), (2), (3), (4)
 - Monitor zadań (1), (2)
 - Zarządzanie przepływem pracy, czynności
 - Harmonogram przetwarzania
 - Skanowanie
 - Skróty, administracja
 - pojedyncze logowanie
 - Lista zadań
 - Przesyłanie
 - ustawienia użytkownika
 - Użytkownicy, grupy, stacje, administracja
 - Przepływy pracy, zadania i czynności, administracja

- [Datacap Navigator, dostęp](#)
- Datacap Navigator, panele niestandardowe
 - nowa lokalizacja w wersji 9.0 pakietu Feature Pack 2 [\(1\)](#), [\(2\)](#)
- Datacap Navigator, układy domyślne
 - nowa lokalizacja w pakiecie Feature Pack 2 [\(1\)](#), [\(2\)](#), [\(3\)](#)
- Datacap, obiekty
 - [API](#)
- Datacap Report Viewer, serwery
 - [Równoważenie obciążenia](#)
- Datacap, raporty
 - [modyfikowanie](#)
- Datacap, serwer
 - Dodawanie listy aplikacji do pliku Datacap.xml
 - [Lista kontrolna](#)
 - Dodawanie lokalizacji pliku Datacap.xml
 - [Lista kontrolna](#)
 - Dodawanie uprawnień do folderu aplikacji
 - [Lista kontrolna](#)
 - Kopiowanie aplikacji
 - [Lista kontrolna](#)
 - Instalowanie
 - [Lista kontrolna](#)
 - Instalowanie i konfigurowanie
 - [Lista kontrolna](#)
 - instalowanie [\(1\)](#), [\(2\)](#)
- Datacap Server
 - [zaawansowane ustawienia bazy danych](#)
 - ustawianie lokalizacji [\(1\)](#), [\(2\)](#)
 - [uruchamianie](#)
- Datacap Server Manager
 - Ustawienia zaawansowane
 - [Akceptowanie połączeń na porcie](#)
 - [Uwierzelnianie, szablon ścieżki](#)
 - [System uwierzelniania](#)
 - [Szablon nazw zadań wsadowych](#)
 - [Limit czasu komendy bazy danych](#)
 - [Blokada okresowa](#)
 - [Metoda uwierzelniania](#)
 - [Limit czasu komendy bazy danych](#)
 - [Datacap, dziennik](#)
 - Datacap Server, usługa [\(1\)](#), [\(2\)](#)
 - [Datacap Server, ustawienia usługi](#)
 - [odmowa dostępu do folderu lub pliku](#)
 - [uwierzelnianie zewnętrzne](#)
 - Uwierzelnianie użytkowników i grup
 - [TMA, uwierzelnianie](#)
 - Uwierzelnianie grupy
 - [ADSI, uwierzelnianie](#)
 - [LDAP, uwierzelnianie](#)
 - LLLDAP, uwierzelnianie grup
 - [Uwierzelnianie, szablon ścieżki](#)
 - [ustawianie w kolejce wg zadań i czynności](#)
 - [Usługa, port połączeń](#)

- [ustawienie rejestrowania zdarzeń](#)
 - [uruchomienie lub zatrzymanie usługi na serwerze](#)
 - [Dziennik zdarzeń systemowych](#)
 - [Użytkownik, uwierzytelnianie](#)
 - [ADLDS, uwierzytelnianie](#)
 - [LLLDAP, uwierzytelnianie](#)
- [Datacap Server, tryb](#)
 - [FastDoc](#)
 - [konfigurowanie aplikacji \(1\), \(2\)](#)
 - [konfigurowanie zestawów reguł](#)
 - [runningFastDoc](#)
 - [uruchamianie FastDoc](#)
- [Datacap Server, właściwości](#)
 - [Przywilej logowania jako usługi](#)
- [Datacap, serwer, usługa](#)
 - [Konfigurowanie](#)
 - [Lista kontrolna](#)
 - [upewnianie się, że konto istnieje](#)
 - [nadawanie uprawnień](#)
- [Datacap Server, usługa](#)
 - [Datacap Server Manager](#)
 - [Ustawienia zaawansowane](#)
 - [Włączenie rejestrowania dla rozwiązywania problemów](#)
- [Datacap Server Service](#)
 - [sprawdzanie statusu](#)
 - [uruchamianie](#)
 - [zatrzymywanie](#)
- [Datacap Server, usługa, ustawienia](#)
 - [Datacap Server Manager](#)
 - [odmowa dostępu do folderu lub pliku](#)
 - [ustawianie w kolejce wg zadań i czynności](#)
 - [ustawienie rejestrowania zdarzeń](#)
 - [uruchomienie lub zatrzymanie usługi na serwerze](#)
- [Datacap, serwery](#)
 - [Równoważenie obciążenia](#)
- [Datacap, oprogramowanie](#)
 - [ponowne uruchamianie](#)
 - [aktualizowanie \(1\), \(2\)](#)
- [Datacap, komponenty oprogramowania](#)
 - [Application Manager](#)
 - [Datacap, bazy danych](#)
 - [Administrator, baza danych](#)
 - [Engine, baza danych](#)
 - [Zewnętrzne, bazy danych](#)
 - [Fingerprint, baza danych](#)
 - [Datacap Server](#)
 - [Datacap Studio](#)
 - [Datacap Web Client](#)
 - [Maintenance Manager](#)
 - [Report Viewer](#)
 - [Rulerunner, usługa](#)
- [Datacap, rozwiązania](#)
- [Datacap, specyficzne parametry](#)

- Datacap, statystyki
 - czynności wykonywane w tle
 - statystki zadań wsadowych
 - statystyki ogólne
 - przegląd
 - czynności skanowania
- Datacap Studio
 - Application Wizard
 - Przekształcanie aplikacji z poprzedniej wersji
 - Kopiowanie aplikacji
 - Tworzenie aplikacji
 - Przypisywanie grupy do zadania wsadowego
 - Dodawanie reguły
 - debugowanie aplikacji
 - opis
 - hierarchia dokumentów
 - testowanie plików dziennika
 - instalowanie komponentów dla programistów (1), (2)
 - otwieranie (1), (2)
 - przetwarzanie zadania wsadowego
 - Rulemanager, karta
 - uruchamianie
 - uruchamianie (1), (2), (3)
 - karty
 - Test, karta
 - interfejs użytkownika
 - Zones, karta
- Datacap Web
 - skróty klawiaturowe
 - aspx, strony WWW
- Datacap Web Client
 - Dodawanie zabezpieczeń
 - Lista kontrolna (1), (2)
 - dodawanie zaufanej strony
 - dodawanie grup
 - Dodawanie zabezpieczeń
 - zaufana strona Opcje internetowe
 - dodawanie stacji
 - Dodawanie użytkowników do Rulerunner
 - AD LDS, uwierzytelnianie
 - LLDAP, uwierzytelnianie
 - Administracja
 - Zadanie wsadowe, tworzenie kolejek użytkowników i stacji
 - Zmiana ustawień SSL
 - server.ini
 - Konfiguracja i testowanie IE
 - Lista kontrolna (1), (2)
 - Konfigurowanie grup i użytkowników
 - Konfigurowanie skrótów
 - Konfigurowanie i testowanie
 - Zdalna stacja robocza
 - Konfigurowanie Internet Explorer
 - Konfigurowanie zabezpieczeń

- zaufana strona Opcje internetowe
 - Konfigurowanie serwisu Web Client Configuration
 - Tworzenie lub upewnianie się, że domena/konto w systemie Windows istnieje
 - Tworzenie czynności skanowania
 - Datacap Navigator
 - Włączenie rejestrowania dla rozwiązywania problemów
 - włączanie rejestrowania
 - Wydruk dokumentu, skanowanie lokalnie
 - Instalowanie
 - Instalowanie i konfigurowanie
 - Monitor zadań (1), (2)
 - logowanie
 - logowanie w celu uwierzytelnienia Rulerunner
 - Zarządzanie przepływem pracy, czynności
 - Harmonogram przetwarzania
 - zdalne skanowanie
 - Zdalna stacja robocza, konfiguracja
 - Lista kontrolna
 - Uruchamianie czynności skanowania
 - skanowanie
 - Ustawianie identyfikatora puli aplikacji
 - Skróty, administracja
 - uruchamianie APT
 - uruchamianie aplikacji Datacap Accounts Payable
 - Testowanie, konfigurowanie
 - Użytkownicy, grupy, stacje, administracja
 - Weryfikowanie zadań wsadowych
 - Weryfikowanie statusu instalacji komponentów IIS
 - Server Manager (1), (2), (3)
 - przeglądanie uprawnień do stacji APT
 - przeglądanie uprawnień użytkownika APT
 - przeglądanie grup uprawnień użytkownika APT
 - Przepływy pracy, zadania i czynności, administracja
- Datacap Web Client Configuration Tool
 - Konfiguracja użytkownika
- Datacap Web Client, serwer
 - kończenie konfiguracji (1), (2)
- Datacap Web Client, serwery
 - Równoważenie obciążenia
- Datacap Web Client, serwis
 - tworzenie (1), (2)
- Datacap Web Client, usługa przesyłania
 - plik konfiguracyjny
 - edytowanie
 - konfigurowanie (1), (2), (3)
 - konfigurowanie aplikacji
 - nawiązywanie połączenia z
 - Datacap Web Client, czynności skanowania
 - wyłączenie
 - instalowanie (1), (2)
 - uruchamianie w postaci usługi
 - przeglądanie dzienników wydarzeń
- Datacap Web Server

- lista kontrolna konfiguracji
- Datacap, usługi WWW
 - zabezpieczenia folderów aplikacji
 - Rozwiązywanie problemów
 - GrabBatch
 - ReleaseBatch
 - SetFile
 - SetPageFileName
 - UploadFile
- Datacap Web Services
 - dodawanie strony WWW wTM
 - ADLDS, uwierzytelnianie
 - ADSI, uwierzytelnianie
 - Application Manager (1), (2)
 - Uwierzytelnianie
 - Użytkownik skonfigurowany
 - Użytkownik końcowy
 - Konfigurowanie uwierzytelniania TMA
 - Application Manager
 - Datacap Windows Service, host
 - Włączenie rejestrowania dla rozwiązywania problemów
 - aktywowanie protokołu SSL
 - GET, metoda
 - CheckIntegrity
 - GetApplicationList
 - GetBatchAttributes
 - GetBatchHistory
 - GetBatchId
 - GetBatchList
 - GetCCO
 - GetFile (1), (2)
 - GetFileList
 - GetGroupPermissionList
 - GetMobileProfiles
 - GetPageFile
 - GetPageFileName
 - GetProgramFile
 - GetUserPermissionList
 - Transakcja/Uruchom
 - Opcje hostingu
 - IIS
 - Windows Service
 - aplikacja IIS, ustawienia pul
 - Instalacja, kroki
 - Instalowanie
 - LDAP, uwierzytelnianie
 - LLDAP, uwierzytelnianie
 - POST, metoda
 - ChangeUserPassword
 - CopyFilesToCache
 - CreateBatch
 - DeleteBatches
 - Wykonywanie (1), (2)

- Wylogowanie
 - Zalogowanie się
 - SaveBatchAttribute
 - SaveTask
 - SetFile (1), (2)
 - SetGroupPermissionList
 - SetUserPermissionList
 - Transaction.End
 - UploadFile
 - PUT, metoda
 - GrabBatch
 - GrabNextPendingBatch OnJobTaskList
 - ReleaseBatch
 - SetPageFileName
 - klucze rejestru
 - API REST, metody
 - punkty końcowe transakcji
- Datacap Web Services, serwery
 - Równoważenie obciążenia
- Datacap Windows Service, host
 - Datacap Web Services
- plik datacap.xml
 - Application Manager (1), (2), (3), (4)
 - ustawienia aplikacji (1), (2), (3), (4)
 - kopiowanie na serwer Datacap
 - kopiowanie na serwer Tasmaster
 - ustawianie lokalizacji (1), (2)
- Datacap.xml, plik
 - Dodawanie listy aplikacji
 - Lista kontrolna
 - Dodawanie odsyłaczy do aplikacji (1), (2)
 - Zmianianie odwołań do aplikacji (1), (2)
 - Serwer Datacap (1), (2), (3)
 - ustawianie lokalizacji
- DatacapBOX, działania
 - AddParentDataToPageMetadata
 - BackupFolder
 - CreateBatchSubfolder
 - DCOVarsAreMetadata
 - DocumentsToPDF
 - Pobieranie
 - FailIfFileExists
 - FieldsAreMetadata
 - ImportAsDocumentType
 - ImportLimit
 - LookforExtensions
 - OverwriteExistingFiles
 - ProcessChildren
 - ReplaceMetadata
 - SourceFolder
 - TargetFolder
 - Przesyłanie
- DatacapBOX Export

- [AddParentDataToPageMetadata](#)
- [BackupFolder](#)
- [CreateBatchSubfolder](#)
- [DCOVarsAreMetadata](#)
- [DocumentsToPDF](#)
- [Pobieranie](#)
- [FailIfFileExists](#)
- [FieldsAreMetadata](#)
- [ImportAsDocumentType](#)
- [LookForExtensions](#)
- [OverwriteExistingFiles](#)
- [ProcessChildren](#)
- [ReplaceMetadata](#)
- [SourceFolder](#)
- [TargetFolder](#)
- [Przesyłanie](#)
- [DatacapBOX, obiekty](#)
 - [Eksport](#)
 - [Import](#)
- [DATAFILE](#)
- [DataType](#)
 - [Konfiguracja DCO](#)
 - [Weryfikacja, panel](#)
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
- [DateStampField, działanie \(1\), \(2\)](#)
- [DB2, baza danych](#)
 - [planowanie](#)
 - [wymagania wstępne](#)
 - [wymagane oprogramowanie](#)
- [DB2, bazy danych](#)
 - [konfigurowanie \(1\), \(2\)](#)
 - [konfigurowanie aplikacja dla](#)
 - [tworzenie](#)
 - [tworzenie pojedynczych użytkowników](#)
 - [planowanie](#)
 - [zapewnianie dostępu](#)
 - [uprawnienia zabezpieczeń](#)
 - [testowanie połączenia](#)
- [Dcclip, działania](#)
 - [dci_clipfield \(1\), \(2\)](#)
- [dci_clipfield, działanie \(1\), \(2\)](#)
- [DCImageFix, działania](#)
 - [ImageEnhance \(1\), \(2\)](#)
 - [LoadSettings \(1\), \(2\)](#)
 - [LoadSettings_FingerprintID \(1\), \(2\)](#)
- [DCO](#)
 - [dodawanie zmiennych liczby całkowitej](#)
 - [AddVariableInt](#)
 - [DCO, metody](#)
 - [dodawanie zmiennych łańcucha](#)
 - [AddVariableString](#)
 - [DCO, metody](#)

- dodawanie zmiennych typu dwustronnego
 - [AddVariableFloat](#)
 - [DCO, metody](#)
- pola wyboru
 - [DCO, metody](#)
 - [get_OMRValue](#)
- sprawdzanie integralności dokumentu
 - [CheckIntegrity](#)
 - [DCO, metody](#)
- obiekt podrzędny, liczba
 - [DCO, metody](#)
 - [NumOfChildren](#)
- ufność, poziomy
 - [DCO, metody](#)
 - [set_CharConfidence](#)
- tworzenie obiektów dokumentów
 - [CreateDocuments](#)
 - [DCO, metody](#)
- usuwanie wartości danych dla znaków
 - [DCO, metody](#)
 - [DeleteValue](#)
- usuwanie obiektów podrzędnych
 - [Czyszczenie](#)
 - [DCO, metody \(1\), \(2\)](#)
 - [DeleteChild](#)
- usuwanie wartości poziomów ufności
 - [DCO, metody](#)
 - [DeleteValue](#)
- usuwanie wartości zmiennych
 - [DCO, metody](#)
 - [DeleteVariable](#)
- usuwanie zmiennych
 - [DCO, metody](#)
 - [DeleteVariable](#)
- pobieranie obiektów podrzędnych
 - [DCO, metody](#)
 - [GetChild](#)
- pobieranie ścieżek do obiektów
 - [DCO, metody](#)
 - [GetRoute](#)
- pobieranie pozycji pól OMR
 - [DCO, metody](#)
 - [get_OMRValue](#)
- pobieranie, wartości zmiennych
 - [DCO, metody](#)
 - [get_Variable](#)
- identyfikowanie typu obiektu
 - [DCO, metody](#)
 - [ObjectType](#)
- identyfikowanie elementów nadrzędnych
 - [DCO, metody](#)
 - [Parent](#)
- przenoszenie obiektów podrzędnych

- DCO, metody (1), (2)
 - [MoveChild](#)
 - [MoveIn](#)
 - wartości danych dla znaków OMR
 - [DCO, metody](#)
 - [set_OMRValue](#)
 - organizowanie obiektów podrzędnych
 - [DCO, metody](#)
 - [MoveChild](#)
 - ustawianie poziomu ufności dla znaków
 - [DCO, metody](#)
 - [set_AltConfidenceString](#)
 - ustawiania wartości dla znaków OMR
 - [pola wyboru](#)
 - [DCO, metody](#)
 - [set_OMRValue](#)
 - ustawianie zmiennych
 - [DCO, metody](#)
 - [set_Variable](#)
 - liczba zmiennych
 - [DCO, metody](#)
 - [NumOfVars](#)
 - zmienne, wartości
 - [DCO, metody](#)
 - [set_Variable](#)
 - writingfiles
 - [DCO, metody](#)
 - [Write](#)
- DCO, działania
 - [ChkConfidence](#) (1), (2)
 - [ChkDCOStatus](#) (1), (2)
 - [ChkDCOType](#) (1), (2)
 - [ChkIntegrity](#) (1), (2)
 - [ChkLastDCOType](#) (1), (2)
 - [ClearAltText](#) (1), (2)
 - [ClearDCO](#) (1), (2)
 - [CopyPD2DD](#) (1), (2)
 - [CountPagesToDocumentVar](#)
 - [CreateDocuments](#) (1), (2)
 - [CreateFields](#) (1), (2)
 - [DeleteFields](#) (1), (2)
 - [IsDocumentCountMoreThan](#)
 - [IsFirstDocumentInBatch](#)
 - [JoinPreviousDocument](#)
 - [PropagateToAltText](#) (1), (2)
 - [RemoveDocumentStructure](#)
 - [SetDCOStatus](#) (1), (2)
 - [SetDCOType](#)
 - [SetDocStatus](#) (1), (2)
 - [SetDocumentType](#) (1), (2)
 - [SetFldConfidence](#) (1), (2)
 - [SetPageFingerprintID](#) (1), (2)
 - [SetPageStatus](#) (1), (2)

- SetPageTemplateID (1), (2)
 - SetPageType (1), (2)
- DCO API
 - tworzenie hierarchii dokumentów
 - Datacap, obiekty
 - DCO, metody
 - AddChild
 - AddValue
 - AddVariable
 - AddVariableFloat
 - AddVariableInt
 - AddVariableString
 - CheckIntegrity
 - Czyszczenie
 - CreateDocuments
 - CreateFields
 - DeleteChild
 - DeleteValue
 - DeleteVariable
 - FindChild
 - FindChildIndex
 - FindRouteChild
 - FindVariable
 - get_AltConfidenceString
 - get_AltText
 - get_CharConfidence
 - get_CharValue
 - get_OMRValue
 - get_Variable
 - GetChild
 - GetLastError
 - GetPosition
 - GetRoute
 - GetVariableName
 - GetVariableValue
 - IsError
 - IsRoute
 - IsValid
 - MoveChild
 - MoveIn
 - NumOfChildren
 - NumOfVars
 - ObjectType
 - Parent
 - Read
 - ReadSetup
 - set_AltConfidenceString
 - set_AltTex
 - set_CharConfidence
 - set_CharValue
 - set_OMRValue
 - SetPosition
 - SetupObject

- Zmienna
 - Write
 - WriteSetup
- DCO, właściwości
 - AltConfidenceString
 - AltText
 - CharConfidence
 - CharValue
 - ConfidenceString
 - ID
 - ImageName
 - Status
 - Tekst
 - Typ
 - Zmienna
 - XML
- DCOSetupNode
- definicja
- składanie dokumentu
- metody
- Właściwości
- Hierarchia zadań wsadowych Runtime
- Środowisko wykonawcze DCO
- Konfiguracja DCO, obiekty podrzędne
- DCO, metody
 - SetupNode
- DCO Tree w Datacap Navigator
- DCOProperty, działanie (1), (2)
- DCOSetup, API
 - DCOSetup, metody
 - AddNode
 - DeleteNode
 - DeleteNodeByName
 - get_DictionaryName
 - get_Value
 - get_Word
 - GetNode
 - GetNodeByName
 - NumOfDictionaries
 - NumOfNodes
 - NumOfWords
 - ReadLock
 - ReadSetup
 - set_DictionaryName
 - set_Value
 - set_Word
 - ShowSetupDialog
 - Unlockit
 - WriteSetup
 - DCOSetup, właściwości
 - DictionaryName
 - Path
 - Value

- Word
 - Hierarchia dokumentów
 - metody
 - właściwości
 - Konfiguracja DCO
- DCOSetupNode, API
 - DCOSetupNode, metody
 - AddRule
 - AddVariable
 - DeleteRule
 - DeleteVariable
 - DeleteVariableByName
 - FindRule
 - get_RuleChildName
 - get_RuleMaxNumber
 - get_RuleMinNumber
 - get_RuleObjectType
 - get_RulePosition
 - get_Variable
 - get_VariableName
 - get_VariableValue
 - GetRule
 - NumOfRules
 - NumOfVariables
 - set_RuleChildName
 - set_RuleMaxNumber
 - set_RuleMinNumber
 - set_RuleObjectType
 - set_RulePosition
 - set_Variable
 - set_VariableName
 - set_VariableValue
 - DCOSetupNode, właściwości
 - Nazwa
 - ObjectType
 - RuleChildName
 - RuleMaxNum
 - RuleMinNum
 - RuleObjectType
 - RulePosition
 - Zmienna
 - VariableName
 - VariableValue
 - metody
 - właściwości
- DCOVarsAreMetadata
- DCOVarsAreMetadata, działanie
- dcpdf, działania
 - dcpdf_CreateTiff FromPDF_CreateDocs
 - dcpdf_CreateTiffFrom PDF_CreateDocs
 - dcpdf_CreateTiffFromPDF (1), (2)
 - dcpdf_MakePDFDoc (1), (2)
 - dcpdf_MaxSizeToReconvert (1), (2)

- [dcpdf_SetApplication \(1\), \(2\)](#)
- [dcpdf_SetAuthor \(1\), \(2\)](#)
- [dcpdf_SetImage Compression](#)
- [dcpdf_SetImageBitcount \(1\), \(2\)](#)
- [dcpdf_SetImageCompression](#)
- [dcpdf_SetImageGrayscale \(1\), \(2\)](#)
- [dcpdf_SetImageQuality \(1\), \(2\)](#)
- [dcpdf_SetImageResolution \(1\), \(2\)](#)
- [dcpdf_SetKeywords \(1\), \(2\)](#)
- [dcpdf_SetProducer \(1\), \(2\)](#)
- [dcpdf_SetSubject \(1\), \(2\)](#)
- [dcpdf_SetTitle \(1\), \(2\)](#)
- [dcpdf_UseAltConversion Method](#)
- [dcpdf_UseAltConversionMethod](#)
- [dcpdf_CreateTiffFrom PDF_CreateDocs, działanie](#)
- [dcpdf_CreateTiffFromPDF, działanie \(1\), \(2\)](#)
- [dcpdf_CreateTiffFromPDF_CreateDocs, działanie](#)
- [dcpdf_MakePDFDoc, działanie \(1\), \(2\)](#)
- [dcpdf_MaxSizeToReconvert, działanie \(1\), \(2\)](#)
- [dcpdf_SetApplication, działanie \(1\), \(2\)](#)
- [dcpdf_SetAuthor, działanie \(1\), \(2\)](#)
- [dcpdf_SetImage Compression, działanie](#)
- [dcpdf_SetImageBitcount, działanie \(1\), \(2\)](#)
- [dcpdf_SetImageCompression, działanie](#)
- [dcpdf_SetImageGrayscale, działanie \(1\), \(2\)](#)
- [dcpdf_SetImageQuality, działanie \(1\), \(2\)](#)
- [dcpdf_SetImageResolution, działanie \(1\), \(2\)](#)
- [dcpdf_SetKeywords, działanie \(1\), \(2\)](#)
- [dcpdf_SetProducer, działanie \(1\), \(2\)](#)
- [dcpdf_SetSubject, działanie \(1\), \(2\)](#)
- [dcpdf_SetTitle, działanie \(1\), \(2\)](#)
- [dcpdf_UseAltConversion Method, działanie](#)
- [dcpdf_UseAltConversionMethod, działanie](#)
- [DD](#)
- [debugowanie aplikacji](#)
- [DebugMode_OFF, działanie \(1\), \(2\)](#)
- [DebugMode_ON, działanie \(1\), \(2\)](#)
- [decyzja, identyfikacja planu](#)
- układy domyślne
 - [Datacap Navigator \(1\), \(2\), \(3\)](#)
- domyślny układ panelu
 - [eksportowanie](#)
- reguły domyślne
 - [przypisywanie do nowych pól](#)
 - [przypisywanie do nowych stron](#)
- [DefaultValue, działanie \(1\), \(2\)](#)
- definiowanie
 - [baza danych, struktury](#)
- [Definiowanie nazw grup w celu filtrowania zadań wsadowych](#)
- [Definiowanie nazw grup w celu filtrowania zadań wsadowych, Application Manager](#)
 - [Application Manager](#)
- [definiowanie, strefy rozpoznawania](#)
- [usuwanie](#)

- [zadania wsadowe](#)
- usuwanie zadań wsadowych
 - [AutoDelete, proces](#)
 - [AutoDelete, zestaw reguł](#)
 - [AutoDelete, program narzędziowy](#)
 - [Datacap Maintenance Manager \(1\), \(2\), \(3\)](#)
- Usuwanie zadań wsadowych
 - POST, metoda
 - [DeleteBatches](#)
 - [Transaction.End](#)
- usuwanie zestawów reguł
 - [Datacap Maintenance Manager, aplikacje](#)
- [DeleteAllAlpha](#), działanie (1), (2)
- [DeleteAllMiscChars](#), działanie (1), (2)
- [DeleteAllNumeric](#), działanie (1), (2)
- [DeleteAllPunct](#), działanie (1), (2)
- [DeleteAllSysChars](#), działanie (1), (2)
- [DeleteChild](#)
 - [DCO, metody](#)
- [DeleteChildType](#), działanie (1), (2)
- [DeleteDirectory](#), działanie (1), (2)
- [DeleteFields](#), działanie (1), (2)
- [DeleteFile](#), działanie (1), (2)
- [DeleteFingerprint](#), działanie
- [DeleteFingerprint](#), działanie (1), (2), (3)
- [DeleteFingerprints](#), działanie (1), (2)
- [DeleteImageFile](#), działanie (1), (2)
- [DeleteLCSpaces](#), działanie (1), (2)
- [DeleteNode](#)
 - [DCOSetup, metody](#)
- [DeleteNodeByName](#)
 - [DCOSetup, metody](#)
- [DeleteParentObj](#), działanie (1), (2)
- [DeleteRule](#)
 - [DCOSetupNode, metody](#)
- [DeleteSelectedChars](#), działanie (1), (2)
- [DeleteValue](#)
 - [DCO, metody](#)
- [DeleteVariable](#)
 - [DCO, metody](#)
 - [DCOSetupNode, metody](#)
- [DeleteVariableByName](#)
 - [DCOSetupNode, metody](#)
- [DeleteWord](#), działanie (1), (2)
- gęstość, wartości łańcuchów
 - [interpretowanie](#)
- [DensityString](#)
- opis
 - [Datacap Accounts Payable](#)
- szczegółowe wymagania systemowe
 - [oprogramowanie, raporty o kompatybilności produktu](#)
- [DetailFix](#), działanie (1), (2)
- programista

- [konto Windows](#)
- stacja robocza programisty
 - [uruchamianie Datacap](#)
- Programista, stacja robocza
 - [Konfigurowanie Internet Explorer](#)
- stacja robocza programisty, komponenty
 - [Datacap Client \(1\), \(2\)](#)
 - [Datacap Studio \(1\), \(2\)](#)
 - [FastDoc \(1\), \(2\)](#)
 - [instalowanie \(1\), \(2\)](#)
 - [Maintenance Manager \(1\), \(2\)](#)
- DICT
 - [Konfiguracja DCO](#)
 - Weryfikacja, panel
 - [Datacap Desktop](#)
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
- słowniki
 - [dotłączanie do pliku](#)
 - [tworzenie](#)
 - [tworzenie w opcjach pól wyboru](#)
- słownik
 - [tworzenie](#)
- Nazwa słownika
 - [ustawienie](#)
- DictionaryName
 - [DCOSetup, właściwości](#)
- [wyłączanie punktów zatrzymania](#)
- [Odłączanie, działanie \(1\), \(2\)](#)
- wykrywanie błędów
 - [DCO, metody](#)
 - [IsError](#)
- [DM_Logon, działanie \(1\), \(2\)](#)
- [DM_SetContentType, działanie](#)
- [DM_SetContentType, działanie](#)
- [DM_SetFolderName, działanie \(1\), \(2\)](#)
- [DM_SetObjectName, działanie \(1\), \(2\)](#)
- [DM_UploadDocument, działanie \(1\), \(2\)](#)
- [DM_UploadPage, działanie \(1\), \(2\)](#)
- Składanie dokumentu
 - [opis](#)
- tworzenie dokumentu i sprawdzanie integralności
 - [przenoszenie do profilu czynności PageID](#)
- hierarchia dokumentów
 - [dodawania kosztów lotu, reguła](#)
 - [dodawanie stron](#)
 - [dodawanie zatwierdzonego typu samochodu, reguła](#)
 - [dodawanie pola zatwierdzonej waluty, reguła](#)
 - [przypisywanie reguł domyślnych](#)
- [Hierarchia dokumentów](#)
 - [API](#)
- Hierarchia dokumentów
 - [aktualizowanie](#)

- hierarchia w dokumencie, kreator aplikacji
 - domyślna
- dokument, indeksowanie
 - przygotowanie dla FastDoc
- dokument, przetwarzanie
 - Formularze, wzorzec
- dokument, skanowanie
 - przygotowanie dla FastDoc
- dokument, ustawienia
 - testowanie
- oddzielanie dokumentu z głównego rozgałęzienia
 - aktualizowanie aplikacji TravelDocs
 - przypisywanie podziału zadań wsadowych, reguła
 - przekazywanie rozdzielonego dokumentu do przełożonego (1), (2), (3), (4)
 - uruchamianie zadania wsadowego w przepływie pracy
 - aktualizowanie przekazywanie, zestaw reguł
- dokument, struktura
 - hierarchia
 - poziomy
- typy dokumentów
 - tworzenie
 - testowanie
- DocumentAnalytics, działania
 - AnalyzeLayout
 - CopyAllBlocks
 - CopyLabelValuePairs
 - CreateCcoFromLayout
 - CreateHTML
 - DocumentAnalytics
 - ExtractText
 - ExtractTextAlchemyLanguage
 - ExtractTextLogEnable
 - FindExtractedText
 - FindLabelValuePair
 - FindLabelValuePairs
 - FindPatterns
- dokumenty
 - składanie
 - sprawdzanie integralności
 - konfigurowanie zestawu reguł w FastDoc
 - konfigurowanie zestawu reguł w FastDoc
 - przekształcanie
 - oparte na hierarchii
 - wstawianie do aplikacji
 - zarządzanie problemami z integralnością
 - przygotowanie do skanowania
 - przetwarzanie we wzorcu uczenia, aplikacje
 - przetwarzanie w FastDoc
 - kierowanie
 - używanie rozgałęzień i dzielenia
 - konfigurowanie w FastDoc (1), (2)
 - określanie struktury
 - oddzielanie z głównego rozgałęzienia

- przypisywanie podziału zadań wsadowych, reguła
 - przekazywanie rozdzielonego dokumentu do przełożonego (1), (2), (3), (4)
 - uruchamianie zadania wsadowego w przepływie pracy
 - aktualizowanie przekazywanie, zestaw reguł
- ustrukturyzowane
- DocumentsToPDF
- DocumentsToPDF, działanie
- Documentum, działania
 - DM_Logon (1), (2)
 - DM_SetContentType (1), (2)
 - DM_SetFolderName (1), (2)
 - DM_SetObjectName (1), (2)
 - DM_UploadDocument (1), (2)
 - DM_UploadPage (1), (2)
- Documentum Connector
 - przesyłanie plików, przykłady
- Documentum Connector, działania
 - konfigurowanie
 - przegląd
 - ustawienia parametrów
 - wymagania wstępne
- DocumentVariable_ExportValue, działanie
- konta domeny
 - tworzenie dla Report Viewer
- Konta domeny
 - Planowanie w systemie Datacap
- nazwa domeny
 - pobieranie do Rulerunner, uwierzytelnianie
- DoMsgbox, działanie (1), (2)
- Statusy pól gotowych
- Statusy stron gotowych
- DotEdit, panele
 - przekształcanie do Datacap Desktop (1), (2), (3)
 - przeglądanie układu, pliki XML (1), (2)
- podwójnie ślepa próba
- Podwójnie ślepa próba
- Pobieranie
- Pobieranie, działanie
- listy rozwijane w Datacap Navigator
- Holenderski
 - kody języków
- dynamiczne ustawienia narodowe, obsługa
 - ustawienia języka
 - zmienne ustawień narodowych
 - nadpisywanie
 - ustawienie
 - przegląd
 - języki rozpoznania

E

- Wschodnioeuropejski
 - kody języków
- dokumentu w formie elektronicznej

- elementy
 - [parametry inteligentne](#)
- Email, działania
 - [SendEMail \(1\), \(2\)](#)
 - [SetAttachment \(1\), \(2\)](#)
 - [SetBlindCarbonCopyRcpts \(1\), \(2\)](#)
 - [SetCarbonCopyRcpts \(1\), \(2\)](#)
 - [SetEmailBody \(1\), \(2\)](#)
 - [SetMailServer \(1\), \(2\)](#)
 - [SetRecipients \(1\), \(2\)](#)
 - [SetSender](#)
 - [SetSubject \(1\), \(2\)](#)
- Email Connector, działania
 - [konfigurowanie](#)
 - [przesyłanie plików, przykłady](#)
 - [przegląd](#)
 - [ustawienia parametrów](#)
 - [wymagania wstępne](#)
- Email Input, działania
 - [opis](#)
- powiadomienia e-mail
 - [wysyłanie](#)
- EmptyFieldValue, działanie [\(1\)](#), [\(2\)](#)
- Włączenie rejestrowania dla rozwiązywania problemów
 - [Datacap Desktop](#)
 - [Datacap Server Service](#)
 - [Datacap Web Client](#)
 - [Datacap Web Services](#)
 - [FastDoc](#)
 - [Rulerunner, usługa](#)
- [EnableEngineLogsOCR_A, działanie \(1\), \(2\)](#)
- [EnableLoggingICR_C, działanie \(1\), \(2\)](#)
- włączanie rejestrowania
 - [Datacap Web Client](#)
- [Włączanie ręcznej rejestracji strony](#)
- klucze szyfrowania
 - [eksportowanie \(1\), \(2\)](#)
 - [generowanie \(1\), \(2\)](#)
 - [generowanie w FastDoc](#)
 - [importowanie \(1\), \(2\), \(3\), \(4\), \(5\), \(6\), \(7\)](#)
 - [rozwiązywanie problemów, niepowodzenie importu](#)
- informacje o końcu świadczenia usług
 - [oprogramowanie, raporty o kompatybilności produktu](#)
- Engine, baza danych
 - [dostęp użytkownika](#)
- angielski
 - [kody języków](#)
- Upewnianie się, że konta istnieją
 - [Użytkownicy](#)
- upewnianie się, że konto istnieje
 - [Datacap, usługa serwisowa](#)
- Equalize, działania
 - [EqualizeUnbalancedImage \(1\), \(2\)](#)

- EqualizeUnbalancedImage, działanie (1), (2)
- EraseRect, działanie
 - opis
- dzienniki zdarzeń
 - [przeglądanie na kliencie Datacap Web Client, przesyłanie, usługa](#)
- Ewsmail, działania
 - czas_przerwania_eksportu (1), (2)
 - gotowy_folder_wyeksportowany (1), (2)
 - eksport_EMLOption (1), (2)
 - wersja_ews_eksportu (1), (2)
 - [limit_czas_HTTP_eksportu](#)
 - [Limit_czas_HTTP_eksportu](#)
 - opcja_właściwości_przesyłania_eksportu (1), (2)
 - logowanie_do_eksportu (1), (2)
 - wylogowywanie_z_eksportu (1), (2)
 - ex_max_docs (1), (2)
 - folder_probleków_eksportu (1), (2)
 - skan_eksportu (1), (2)
 - typy_eksportu (1), (2)
 - czas_oczekiwania_na_eksport (1), (2)
- czas_przerwania_eksportu, działanie (1), (2)
- gotowy_folder_wyeksportowany, działanie (1), (2)
- eksport_EMLOption, działanie (1), (2)
- wersja_ews_eksportu, działanie (1), (2)
- limit_czas_HTTP_eksportu, działanie (1), (2)
- opcja_właściwości_przesyłania_eksportu, działanie (1), (2)
- logowanie_do_eksportu, działanie (1), (2)
- wylogowywanie_z_eksportu, działanie (1), (2)
- ex_max_docs, działanie (1), (2)
- folder_probleków_eksportu, działanie (1), (2)
- skan_eksportu, działanie (1), (2)
- typy_eksportu, działanie (1), (2)
- czas_oczekiwania_na_eksport, działanie (1), (2)
- przykłady
 - przystanie plików
 - [Box Connector](#)
 - [Documentum Connector](#)
 - [Email Connector, działania](#)
 - [FileNet Image Services Connector](#)
 - [FileNet P8 Connector](#)
 - [IBM Content Manager Connector](#)
 - [SharePoint Connector](#)
 - [import z faksu](#)
- Excel, działania
 - ExcelAutoFitColumns (1), (2)
 - ExcelAutoFitRows (1), (2)
 - ExcelOrientationToLandscape (1), (2)
 - ExcelOrientationToPortrait (1), (2)
 - ExcelPrintBlankPage (1), (2)
 - ExcelPrintGridlines (1), (2)
 - ExcelPrintQuality (1), (2)
 - ExcelScalingFactor (1), (2)
 - ExcelTiffCompression (1), (2)

- [ExcelWorkbookToImage \(1\), \(2\)](#)
 - [ExcelWorkbookToImageEx](#)
- [ExcelAutoFitColumns](#), działanie (1), (2)
- [ExcelAutoFitRows](#), działanie
- [ExcelAutoFitRows](#), działanie
- [ExcelOrientationToLandscape](#), działanie (1), (2)
- [ExcelOrientationToPortrait](#), działanie (1), (2)
- [ExcelPrintBlankPage](#), działanie (1), (2)
- [ExcelPrintGridlines](#), działanie (1), (2)
- [ExcelPrintQuality](#), działanie (1), (2)
- [ExcelScalingFactor](#), działanie (1), (2)
- [ExcelTiffCompression](#), działanie (1), (2)
- [ExcelWorkbookToImage](#), działanie (1), (2)
- [ExcelWorkbookToImageEx](#), działanie
- [ExceptionSetFileTypes](#), działanie (1), (2)
- [ExceptionSetHandler](#), działanie (1), (2)
- [ExceptionSetTaskCondition](#), działanie (1), (2)
- [ExceptionSetVariableName](#), działanie (1), (2)
- Wykonywanie reguł na zadaniu wsadowym
 - POST, metoda
 - Wykonywanie (1), (2)
- [ExecuteSQL](#), działanie (1), (2)
- [ExecuteSQLBind](#), działanie (1), (2)
- eksport
 - klucze szyfrowania (1), (2)
- [Eksport](#)
- Eksport, działania
 - [BatchVariable_ExportValue \(1\), \(2\)](#)
 - [BlankFields \(1\), \(2\)](#)
 - [BlankLines \(1\), \(2\)](#)
 - [BPilot \(1\), \(2\)](#)
 - [CloseExportFile \(1\), \(2\)](#)
 - [DCOProperty \(1\), \(2\)](#)
 - [DocumentVariable_Export Value](#)
 - [DocumentVariable_ExportValue](#)
 - [ExportAllFields \(1\), \(2\)](#)
 - [ExportFieldValue \(1\), \(2\)](#)
 - [ExportMYValue \(1\), \(2\)](#)
 - [ExportSmartParameter \(1\), \(2\)](#)
 - [ExportToBatchDir \(1\), \(2\)](#)
 - [Filler \(1\), \(2\)](#)
 - [FixedLenLJ \(1\), \(2\)](#)
 - [FixedLenRJ \(1\), \(2\)](#)
 - [GetDATE \(1\), \(2\)](#)
 - [GetProfileString \(1\), \(2\)](#)
 - [GetTime \(1\), \(2\)](#)
 - [LineItem_AddElement \(1\), \(2\)](#)
 - [LineItem_BlankFields \(1\), \(2\)](#)
 - [LineItem_ClearElements \(1\), \(2\)](#)
 - [LineItem_ExportElements \(1\), \(2\)](#)
 - [LineItem_SmartParameter \(1\), \(2\)](#)
 - [NewLine \(1\), \(2\)](#)
 - [PageVariable_ExportValue \(1\), \(2\)](#)

- ResetFieldVariables (1), (2)
- SaveFilePathAsVariable (1), (2)
- SetCSV (1), (2)
- SetElementSeparator (1), (2)
- SetExportPath (1), (2)
- SetExtensionName (1), (2)
- SetFileName (1), (2)
- SetFill (1), (2)
- SetFixedLength (1), (2)
- SetIgnoreFieldStatus (1), (2)
- SetJustified (1), (2)
- SetOMR_Separator (1), (2)
- SetSpaceFill (1), (2)
- SetZeroFill (1), (2)
- Text (1), (2)
- Variable_ExportValue (1), (2)
- Variable_IsValue (1), (2)
- eksport, baza danych
 - [konfigurowanie](#)
- eksport, tabla bazy danych
 - [tworzenie](#)
- Eksportowanie danych dotyczących umowy wynajmu, reguła
 - [dołączanie do strony z umową wynajmu](#)
- eksport, konfigurowanie zestawu reguł
 - [FastDoc](#)
- eksport, czynność
 - testowanie (1), (2)
- Eksport, profil czynności
 - dodawanie zestawu reguł
 - [ExportDB, zestaw reguł](#)
 - dodawanie zestawu reguł
 - [Export XML, zestaw reguł](#)
- [Eksport do formatu pliku Folderdata](#)
- Export XML, reguły
 - [dołączanie do hierarchii dokumentu](#)
- ExportAllFields, działanie (1), (2)
- ExportBatchIDToColumn, działanie (1), (2)
- ExportCloseConnection, działanie (1), (2)
- ExportDB, działania
 - AddRecord (1), (2)
 - ExportBatchIDToColumn (1), (2)
 - ExportCloseConnection (1), (2)
 - ExportFieldToColumn (1), (2)
 - ExportNodeXMLToColumn (1), (2)
 - ExportOpenConnection (1), (2)
 - ExportPropertyToColumn (1), (2)
 - [ExportSmartParam ToColumn](#)
 - [ExportSmartParamToColumn](#)
 - ExportToColumn (1), (2)
 - SetTableName (1), (2)
- ExportDB, zestaw reguł
 - [dodawanie reguł](#)
 - [tworzenie](#)

- [wyeksportowany plik danych](#)
- [ExportFieldToColumn](#), działanie (1), (2)
- [ExportFieldValue](#), działanie (1), (2)
- [Eksportowanie informacji z DCO](#)
- [eksportowanie informacji o pozycji](#)
- eksportowanie tekstu
 - [IBM Content Manager OnDemand](#)
- eksportowanie do XML
 - [tworzenie zestawu reguł](#)
- [ExportMYValue](#), działanie (1), (2)
- [ExportNodeXMLToColumn](#), działanie (1), (2)
- [ExportOpenConnection](#), działanie (1), (2)
- [ExportPropertyToColumn](#), działanie (1), (2)
- [ExportSmartParameter](#), działanie (1), (2)
- [ExportSmartParamToColumn](#), działanie
- [ExportToBatchDir](#), działanie (1), (2)
- [ExportToColumn](#), działanie (1), (2)
- [ExportXML](#), działania
 - [xml_CommitNode](#) (1), (2)
 - [xml_NewNode](#) (1), (2)
 - [xml_SaveFile](#) (1), (2)
 - [xml_SetAttributeValue](#) (1), (2)
 - [xml_SetExportPath](#) (1), (2)
 - [xml_SetFileName](#) (1), (2)
 - [xml_SetNodeValue](#) (1), (2)
- [ExportXML](#), zestaw reguł
 - [uzyskiwanie dostępu do hierarchii środowiska wykonawczego](#) (1), (2)
 - [tworzenie](#)
- wyrażenia
 - [budowanie](#)
- Dane zewnętrzne, usługi w Datacap Navigator (1), (2)
- [ExtractText](#), działanie
- [ExtractTextAlchemyLanguage](#), działanie
- [ExtractTextLogEnable](#), działanie

F

- [FailIfFileExists](#)
- [FailIfFileExists](#), działanie
- [FailRuleSet](#), działanie (1), (2)
- [FastDoc](#)
 - [tworzenie kopii zapasowe, aplikacje](#)
 - [przechwytywanie, dane indeksu](#)
 - [konfigurowanie](#) (1), (2)
 - [konfigurowanie zestawów reguł eksportu](#)
 - [konfigurowanie zestawów reguł](#)
 - [potwierdzanie eksportu](#)
 - [tworzenie zadań wsadowych](#)
 - [za pomocą skanera](#)
 - [przy użyciu zeskanowanych obrazów](#)
 - [tworzenie odcisków](#)
 - [Datacap Server](#), tryb (1), (2)
 - [konfigurowanie aplikacji](#) (1), (2)
 - [konfigurowanie zestawów reguł](#)

- definiowanie indeksu, uzupełnianie pól
 - korzystanie z bazy danych
- definiowanie indeksu, sprawdzanie poprawności pól
 - korzystanie z bazy danych
- definiowanie indeksu, pola
 - korzystając ze słów kluczowych
- usuwanie zadań wsadowych (1), (2)
- usuwanie odcisków
- usuwanie stron
- opis
- określanie nazw plików
- wyświetlanie dokumentów
- Włączenie rejestrowania dla rozwiązywania problemów
- wyeksportowane dane, format pliku
- eksportowanie obrazów
- Formularze, konfiguracja wzorców
- generowanie kluczy zabezpieczeń
- indeksowanie zadań wsadowych
- instalowanie (1), (2)
- instalowanie komponentów dla programistów (1), (2)
- instalowanie na jednym komputerze
- skróty klawiaturowe
- Uczenie, konfiguracja wzorców
- Lokalny, tryb (1), (2)
 - konfigurowanie aplikacji
 - konfigurowanie profili zadań wsadowych
- zarządzanie plikami (1), (2), (3)
- otwieranie zadań wsadowych
- organizowanie stron w dokumencie
- przygotowywanie
- przygotowanie dokumentów do skanowania
- przygotowanie do
 - indeksowanie
 - skanowanie
- wymagania wstępne
- przetwarzanie dokumentów
- oczyszczanie, zadania wsadowe (1), (2)
- szybkie tworzenie aplikacji
- uruchamianie
- uruchamianie w trybie Datacap Server
- uruchamianie w trybie lokalnym
- skanowanie zeskanowanych obrazów
- wybieranie języka systemu Windows
- konfigurowanie skanera
- konfigurowanie dokumentów (1), (2)
- konfigurowanie pól
- instalacja autonomiczna (1), (2)
- uruchamianie
- testowanie, ustawienia
- rozwiązywanie problemów
 - słabe rozpoznawanie kodów kreskowych
 - typ dokumentu nie jest przypisywany automatycznie
 - błąd podczas eksportu

- dodatkowe dane w danych przechwytywania
 - dane indeksowe niewybrane w polu strefy
 - skaner nie znajduje się na liście
 - SharePoint, przesyłanie (1), (2), (3)
 - przesyłanie zadań wsadowych
 - korzystanie z klucza klikania N
 - korzystanie ze aplikacji wzorca uczenia
 - korzystanie ze wzorca formularzy (1), (2)
 - korzystanie ze wzorca uczenia
 - weryfikowanie zadań wsadowych
- FastDoc, przeglądanie statystyk dziennych
- Faks, działania
 - uwierzytelnianie
- Fax Connector, działania
 - konfigurowanie
 - ustawienia parametrów
 - wymagania wstępne
- Fax Connector, działania
 - import, przykłady
- Opinia, działanie (1), (2)
- pole, dane
 - lokalizowanie za pomocą dopasowywania tekstu
- pole, definicje
 - współużytkowanie w zakresie hierarchii dokumentu
- pole, nazwy
 - zmiana
- pole, pozycja
 - DCO, metody (1), (2)
 - GetPosition
 - SetPosition
- pole, wartości
 - określanie
- FieldContainsValue, działanie (1), (2)
- pola
 - weryfikacja danych
 - na stronach
 - określanie struktury
 - zapobieganie nadpisywaniu niepowodzeń w sprawdzaniu poprawności
 - konfigurowanie we wzorcu uczenia
- FieldsAreMetadata
- FieldsAreMetadata, działanie
- plik, zarządzanie
 - FastDoc (1), (2), (3)
- FileIO, działania
 - AppendAllImages
 - AppendAllImages_ByType
 - AppendImage
 - AppendImage_StartAsNew
 - CheckFreeDiskSpace
 - CheckFreeSpace
 - ConvertToJPEG
 - ConvertToTIFF
 - CopyDirectory (1), (2)

- CopyFile (1), (2)
- DeleteDirectory (1), (2)
- DeleteFile (1), (2)
- GetFileSize (1), (2)
- GetProfileString (1), (2)
- IsDirectoryPresent (1), (2)
- IsFilePresent (1), (2)
- IsFileReadOnly (1), (2)
- IsProfilePresent (1), (2)
- RenameFile (1), (2)
- SetChrominanceFactor
- SetDeleteOriginal
- SetFileReadOnly (1), (2)
- SetGrayScale
- SetLuminanceFactor
- SetProfileString (1), (2)
- SetTIFFCompression
- SplitFileName (1), (2)
- FileNet Doc ID Set Value, działanie
- FileNet Image Services Connector
 - przesyłanie plików, przykłady
- FileNet Image Services Connector, działania
 - konfigurowanie
 - przegląd
 - ustawienia parametrów
 - wymagania wstępne
- FileNet P8, działania
 - ExcelWorkbookToPdf
 - FNP8_SetDestinationFolder
 - FNP8_SetDocClassId
 - FNP8_SetDocTitle
 - FNP8_SetFileType
 - FNP8_SetKeyProperty
 - FNP8_SetLocale
 - FNP8_SetMimeType
 - FNP8_SetMultiValueProperty
 - FNP8_SetProperty
 - FNP8_SetRetry
 - FNP8_SetTargetClassID
 - FNP8_SetTargetObjectID
 - FNP8_SetTimeout
 - FNP8_SetUploadMode
 - FNP8_SetURL
 - FNP8_CreateFolder (1), (2)
 - FNP8_Login (1), (2)
 - FNP8_MultiPageDocs
 - FNP8_SetDestinationFolder
 - FNP8_SetDocClassId
 - FNP8_SetDocTitle
 - FNP8_SetFileMimeType
 - FNP8_SetFileType
 - FNP8_SetKeyProperty
 - FNP8_SetLocale

- [FNP8_SetMultiValue Property](#)
- [FNP8_SetProperty](#)
- [FNP8_SetPropertyEx](#)
- [FNP8_SetRetry](#)
- [FNP8_SetTargetClassID](#)
- [FNP8_SetTargetObjectID](#)
- [FNP8_SetTimeout](#)
- [FNP8_SetUploadMode](#)
- [FNP8_SetURL](#)
- [FNP8_UpdateProperties \(1\), \(2\)](#)
- [FNP8_Upload \(1\), \(2\)](#)
- [FNP8_UploadDir \(1\), \(2\)](#)
- [HtmlToPdf](#)
- [standardy zgodności PDF](#)
- [typy kompresji obrazów PDF](#)
- [PDFFREDocumentToImage](#)
- [RtfToPdf](#)
- [TxtToPdf](#)
- [FileNet P8 Connector](#)
 - [przesyłanie plików, przykłady](#)
- [FileNet P8 Connector, działania](#)
 - [konfigurowanie \(1\), \(2\)](#)
 - [ustawienia parametrów](#)
 - [wymagania wstępne](#)
- [FileNetDB_ADOConnect, działanie \(1\), \(2\)](#)
- [FileNetDocID_SaveAsSmartParameter, działanie](#)
- [FileNetDocID_SetValue, działanie](#)
- [FileNETDocID_SetValue, działanie](#)
- [FileNetIDM](#)
 - [Uwierzytelnianie](#)
- [FileNetIDM, działania](#)
 - [AddAllImagesToDocument \(1\), \(2\)](#)
 - [AddFileToDocument \(1\), \(2\)](#)
 - [AddPDFImageToDocument \(1\), \(2\)](#)
 - [AddTIFFImageToDocument \(1\), \(2\)](#)
 - [CreateFolder \(1\), \(2\)](#)
 - [FileNet Doc ID Set Value](#)
 - [FileNetDB_ADOConnect \(1\), \(2\)](#)
 - [FileNetDocID_SaveAsSmartParameter](#)
 - [FileNETDocID_SaveAs SmartParameter](#)
 - [FileNetDocID_SetValue](#)
 - [FileNETDocID_SetValue](#)
 - [GetDocuments \(1\), \(2\)](#)
 - [GetTopFolders \(1\), \(2\)](#)
 - [IndexProperty_ID_DateComponent](#)
 - [IndexProperty_ID_Component \(1\), \(2\)](#)
 - [IndexProperty_ID_DateComponent](#)
 - [IndexProperty_ID_Value](#)
 - [IndexProperty_LeftJUSTIFY \(1\), \(2\)](#)
 - [IndexProperty_RightJUSTIFY \(1\), \(2\)](#)
 - [IndexProperty_SmartParameter \(1\), \(2\)](#)
 - [IndexProperty_ID_Value](#)
 - [Library_DMA_Initialize \(1\), \(2\)](#)

- Library_DS_Initialize (1), (2)
- Library_IS_Initialize (1), (2)
- Library_LogIn (1), (2)
- Library_LogOff (1), (2)
- NewDocument (1), (2)
- SaveDocToFolder (1), (2)
- Upload (1), (2)
- Upload_SetNumAttempts (1), (2)
- UploadSetDelay
- UseIndexes_ON
- UseIndexes_OFF (1), (2)
- UseIndexes_ON
- Filler, działanie (1), (2)
- Filtrowanie zadań wsadowych według grupy
 - Uwierzytelnianie grupy
 - ADSI, uwierzytelnianie grup
 - LDAP, uwierzytelnianie grup
 - LLDAP, uwierzytelnianie grup
 - Monitor zadań
- FilterFieldSelectedChars, działanie (1), (2)
- FilterIt, działanie (1), (2)
- FilterPID, działanie
- filtry
 - konfigurowanie
- FindBlackFingerprint, działanie (1), (2)
- FindBlocks_WhiteSpace, działanie (1), (2)
- FindChild
 - DCO, metody
- FindChildIndex
 - DCO, metody
- FindDataBlocks, działanie (1), (2)
- FindDBList, działanie (1), (2)
- FindDBList_InZone, działanie (1), (2)
- FindExportImage, działanie (1), (2)
- FindExtractedText, działanie
- FindFields, działanie
- FindFingerprint, działanie (1), (2)
- FindFingerprintCC, działanie
- wyszukiwanie, obiekt podrzędny, indeks
 - DCO, metody
 - FindChildIndex
- wyszukiwanie, obiekty podrzędne
 - DCO, metody
 - FindChild
- wyszukiwanie, interfejs obiektów
 - DCO, metody
 - FindRouteChild
- wyszukiwanie, zmienne, indeks
 - DCO, metody
 - FindVariable
- FindKeyList, działanie (1), (2)
- FindKeyList_InZone, działanie (1), (2)
- FindLabelValuePair, działanie

- [FindLabelValuePairs](#), działanie
- FindLastKeyList, działanie (1), (2)
- FindLastKeyList_InZone, działanie (1), (2)
- FindLastRegEx, działanie (1), (2)
- FindLastRegEx_InZone, działanie (1), (2)
- FindLastRegExList, działanie (1), (2)
- FindLastRegExList_InZone, działanie (1), (2)
- FindLastWord, działanie (1), (2)
- FindLastWord_InZone, działanie (1), (2)
- FindLineItems, działanie (1), (2)
- FindNextDBList, działanie (1), (2)
- FindNextDBList_InZone, działanie (1), (2)
- FindNextKeyList, działanie (1), (2)
- FindNextKeyList_InZone, działanie (1), (2)
- FindNextRegExList, działanie (1), (2)
- FindNextRegExList_InZone, działanie (1), (2)
- [FindPatterns](#), działanie
- FindRegExBlocks, działanie (1), (2)
- FindRegExList, działanie (1), (2)
- FindRegExList_InZone, działanie (1), (2)
- FindRouteChild
 - [DCO](#), metody
- FindRule
 - [DCOSetupNode](#), metody
- FindTaxValue, działanie
 - opis
- FindTemplate, działanie (1), (2)
- FindVariable
 - [DCO](#), metody
- FindZoneLineItems, działanie (1), (2)
- [odcisk](#)
 - [wdrażanie nowych ustawień przetwarzania obrazów](#)
 - [zmienianie metod tworzenia](#)
 - [tworzenie klas](#)
 - [udoskonalanie obrazów](#)
- [Utworzono odcisk](#)
- baza danych odcisków
 - opis
- Baza danych odcisków
 - [dostęp użytkownika](#)
- pliki odcisków
 - [tworzenie](#)
 - [nadawanie uprawnień do](#)
- [Odcisk, funkcje](#)
- odcisk, generowanie
 - [automatyczne](#)
- biblioteka odcisków
 - [dodawanie odcisków](#)
 - [tworzenie inicjałów dla TravelDocs](#)
- Fingerprint Maintenance Tool
 - [dodawanie odcisków](#)
 - [przyciski](#)
 - [konfiguracja](#)

- usuwanie odcisków
 - powiązane z typami dokumentów
 - usuwanie częściowych odcisków
 - eksportowanie odcisków
 - pola
 - zarządzanie odciskami
 - odwołanie
 - Settings.ini
 - konfiguracja
 - uruchamianie
 - rozwiązywanie problemów
 - katalog kopii zapasowej
 - plik FMT.log
 - korzystanie z (1), (2)
- Fingerprint Management
- odcisk, ustalanie zgodności
 - tryby tworzenia
 - cała strona, rozpoznanie
 - analiza obrazu
- Odcisk, usługa
 - Konfigurowanie
 - Lista kontrolna
 - Testowanie instalacji
 - Lista kontrolna
- Fingerprint, usługa
 - Uprawnienia dostępu do konta
 - c:\Datacap\application\batches
 - c:\Datacap\application\fingerprint
 - konta
 - Działania
 - SetApplicationID
 - Dodawanie konta do grupy IIS_IUSRS
 - Dodawanie puli aplikacji
 - instalowanie i konfigurowanie
 - instalacja
 - Microsoft Internet Information Services
 - uprawnienia
 - wymagania wstępne
 - zabezpieczenia
 - FingerprintService, folder
 - weryfikowanie (1), (2)
- Fingerprint Service, serwery
 - Równoważenie obciążenia
- Fingerprint, plik XML
 - FPXML
 - Strefa, pozycja
- pliki XML odcisków
- FingerprintMaintenance, działania
 - CloseDatabase (1), (2)
 - DeleteFingerprint (1), (2)
 - DeleteFingerprints (1), (2)
 - OpenDatabase (1), (2)
 - SetFingerprintFolder (1), (2)

- odciski
 - dodawanie
 - korzystanie z działań
 - korzystanie z karty Datacap Studio Zones
 - tworzenie
 - tworzenie na typy stron
 - tworzenie w FastDoc (1), (2)
 - tworzenie, strefy rozpoznania
 - definiowanie stref pól
 - usuwanie (1), (2)
 - usuwanie w FastDoc
 - włączanie FPXML
 - eksportowanie
 - generowanie automatyczne (1), (2)
 - zarządzanie
 - uruchamianie narzędzia Fingerprint Maintenance Tool
 - zapisywanie (1), (2)
 - synchronizowanie
- Fix Central
 - pobieranie poprawek za pomocą
- FixedLenLJ, działanie (1), (2)
- FixedLenRJ, działanie (1), (2)
- poprawki
 - pobieranie
 - Fix Central
- Poprawka, czynność
 - Medical Claims
- Kosztów lotów, reguła
 - dodawanie do hierarchii dokumentu
- FNP8_SetDestinationFolder, działanie
- FNP8_SetDocClassId, działanie
- FNP8_SetDocTitle, działanie
- FNP8_SetFileType, działanie
- FNP8_SetKeyProperty, działanie
- FNP8_SetLocale, działanie
- FNP8_SetMimeType, działanie
- FNP8_SetMultiValueProperty, działanie
- FNP8_SetProperty, działanie
- FNP8_SetRetry, działanie (1), (2)
- FNP8_SetTargetClassID, działanie
- FNP8_SetTargetObjectID, działanie
- FNP8_SetTimeout, działanie
- FNP8_SetUpdateProperties, działanie
- FNP8_SetUploadMode, działanie
- FNP8_SetURL, działanie
- FNP8_CreateFolder, działanie (1), (2)
- FNP8_Login, działanie (1), (2)
- FNP8_MultiPageDocs, działanie
- FNP8_SetDestinationFolder, działanie
- FNP8_SetDocClassId, działanie
- FNP8_SetDocTitle, działanie
- FNP8_SetFileType, działanie
- FNP8_SetKeyProperty, działanie

- [FNP8_SetLocale](#), działanie
- właściwość [FNP8_SetMultiValue](#), działanie
- [FNP8_SetProperty](#), działanie
- [FNP8_SetPropertyEx](#), działanie
- [FNP8_SetTargetClassID](#), działanie
- [FNP8_SetTargetObjectID](#), działanie
- [FNP8_SetTimeout](#), działanie
- [FNP8_SetUploadMode](#), działanie
- [FNP8_SetURL](#), działanie
- [FNP8_UpdateProperties](#), działanie
- [FNP8_Upload](#), działanie (1), (2)
- [FNP8_UploadDir](#), działanie (1), (2)
- stopka, wiersze
 - [dodawanie](#)
- [FormatFieldLengths](#), działanie
- [FormatNumberToLocale](#), działanie (1), (2)
- Formularze, wzorzec
 - [czyszczenie obrazów](#)
 - [konfigurowanie w FastDoc](#)
 - [zadania](#)
 - [przetwarzanie dokumentów](#)
 - [konfigurowanie dokumentów w FastDoc](#)
 - [konfigurowanie rozpoznawania pól](#)
 - [konfigurowanie sprawdzania poprawności pól](#)
- FPXML
 - [automatyczne odczytywanie odcisków](#)
 - [włączanie](#)
 - Fingerprint, plik XML
 - [Strefa, pozycja](#)
- FPXML, działania
 - [ReadZonesFPX](#) (1), (2)
 - [SetDetailsAndLineitemPairFPX](#) (1), (2)
 - [SetDirectoryFPX](#) (1), (2)
 - [WriteZoneFPX](#) (1), (2)
 - [WriteZonesFPX](#) (1), (2)
- [FPXMLUsed](#), działanie (1), (2)
- Francuski
 - [kody języków](#)
- Pełne punkty zatrzymania

G

- generowanie
 - układ, pliki XML (1), (2)
 - [raport, dane](#)
- [GenerateDetails](#), działanie (1), (2)
- generowanie
 - [automatyczne generowanie odcisku](#)
 - [odciski](#)
- ogólne punkty zatrzymania
 - [ustawienie](#)
- [geokodowanie i lokalizacja](#)
- dopasowywania wzorca geometrycznego
 - [dopasowywanie wzorca](#)

- aktualizowanie aplikacji TravelDocs
 - przeglądanie plików zadań wsadowych w środowisku wykonawczym
 - uruchamianie zadania wsadowego w przepływie pracy
 - konfigurowanie strefy zakotwiczenia
 - aktualizowanie PageID, reguła
 - korzystanie
- Niemiecki
 - kody języków
- get_AltConfidenceString
 - DCO, metody
- get_AltText
 - DCO, metody
- Pobieranie, lista zadań wsadowych
 - GET, metoda
 - GetBatchList
- Pozyskaj informacje o CCO
 - GET, metoda
 - GetCCO
- get_CharConfidence
 - DCO, metody
- get_CharValue
 - DCO, metody
- get_DictionaryName
 - DCOSetup, metody
- Pobieranie, lista plików
 - GET, metoda
 - GetFileList
- GET, metoda
 - CheckIntegrity
 - GetApplicationList
 - GetBatchAttributes
 - GetBatchHistory
 - GetBatchId
 - GetBatchList
 - GetCCO
 - GetFile (1), (2)
 - GetFileList
 - GetGroupPermissionList
 - GetMobileProfiles
 - GetPageFile
 - GetPageFileName
 - GetProgramFile
 - GetUserPermissionList
 - Transakcja/Uruchom
- get_OMRValue
 - DCO, metody
- Pobieranie, treści z pliku strony
 - GET, metoda
 - GetPageFile
- Pobieranie, nazwa pliku strony
 - GET, metoda
 - GetPageFileName
- get_RuleChildName

- [DCOSetupNode, metody](#)
- [get_RuleMaxNumber](#)
 - [DCOSetupNode, metody](#)
- [get_RuleMinNumber](#)
 - [DCOSetupNode, metody](#)
- [get_RuleObjectType](#)
 - [DCOSetupNode, metody](#)
- [get_RulePosition](#)
 - [DCOSetupNode, metody](#)
- [get_Value](#)
 - [DCOSetup, metody](#)
- [get_Variable](#)
 - [DCO, metody](#)
 - [DCOSetupNode, metody](#)
- [get_VariableName](#)
 - [DCOSetupNode, metody](#)
- [get_VariableValue](#)
 - [DCOSetupNode, metody](#)
- [get_Word](#)
 - [DCOSetup, metody](#)
- [Get2DCodesBP, działanie \(1\), \(2\)](#)
- [GetAllBarcodes, działanie](#)
 - [opis](#)
- [GetAllBarcodesBP, działanie \(1\), \(2\)](#)
- [GetBarCode, działanie \(1\), \(2\)](#)
- [GetBarcodeBP, działanie](#)
- [GetBarcodesBP, działanie](#)
- [GetChild](#)
 - [DCO, metody](#)
- [GetDataMatrixCodeBP, działanie \(1\), \(2\)](#)
- [GetDATE, działanie \(1\), \(2\)](#)
- [GetDocuments, działanie \(1\), \(2\)](#)
- [GetFileSize, działanie \(1\), \(2\)](#)
- [GetJobID, działanie \(1\), \(2\)](#)
- [GetLastError](#)
 - [DCO, metody](#)
- [GetNode](#)
 - [DCOSetup, metody](#)
- [GetNodeByName](#)
 - [DCOSetup, metody](#)
- [GetPosition](#)
 - [DCO, metody](#)
- [GetProfileString, działanie \(1\), \(2\), \(3\), \(4\)](#)
- [GetRoute](#)
 - [DCO, metody](#)
- [GetRule](#)
 - [DCOSetupNode, metody](#)
- [GetSelectedBlockType, działanie](#)
- [GetTime, działanie \(1\), \(2\)](#)
- pobieranie, dane pól znaków alternatywnych
 - [AltText](#)
 - [DCO, właściwości](#)
- pobieranie znaków, wartości danych ASCII

- [DCO, metody](#)
 - [get_CharValue](#)
- pobieranie znaków, dane
 - [DCO, metody \(1\), \(2\)](#)
 - [DCO, właściwości](#)
 - [get_AltText](#)
 - [get_CharConfidence](#)
 - [Tekst](#)
- pobieranie znaków, wartość ufności dla danych
 - [ConfidenceString](#)
 - [DCO, właściwości](#)
- pobieranie znaków, wartość danych
 - [CharValue](#)
 - [DCO, właściwości](#)
- pobieranie znaków, poziom ufności dla wartości
 - [CharConfidence](#)
 - [DCO, właściwości](#)
- pobieranie, obiekt podrzędny, interfejs
 - [DCO, metody](#)
 - [FindChild](#)
- pobieranie, poziomu ufności dla znaków
 - [AltConfidenceString](#)
 - [DCO, metody](#)
 - [DCO, właściwości](#)
 - [get_AltConfidenceString](#)
- pobieranie, nazwa pliku obrazu
 - [DCO, właściwości](#)
 - [ImageName](#)
- pobieranie, ostatni błąd
 - [DCO, metody](#)
 - [GetLastError](#)
- pobieranie, nazwa obiektu
 - [DCO, właściwości](#)
 - [Typ](#)
- pobieranie, unikalny identyfikator obiektu
 - [DCO, właściwości](#)
 - [ID](#)
- pobieranie, właściwość status
 - [DCO, właściwości](#)
 - [Status](#)
- pobieranie, zmienna
 - [DCO, właściwości](#)
 - [Zmienna](#)
- pobieranie, nazwy zmiennych
 - [DCO, metody](#)
 - [GetVariableName](#)
- pobieranie, wartości zmiennych
 - [DCO, metody](#)
 - [GetVariableValue](#)
- pobieranie, nazwa pliku XML
 - [DCO, właściwości](#)
 - [XML](#)
- [GetTopFolders](#), działanie (1), (2)

- [GetVariableName](#)
 - [DCO, metody](#)
- [GetVariableValue](#)
 - [DCO, metody](#)
- [GetZoneText, działanie \(1\), \(2\)](#)
- [działania globalne](#)
 - [Aplikacja, konfiguracja](#)
 - [Autodoc](#)
 - [Kod_kreskowy_P](#)
 - [Kod_kreskowy_X](#)
 - [Przetwarzanie wsadowe](#)
 - [CC](#)
 - [Cco2cco](#)
 - [sprawdzanie przetwarzania \(1\), \(2\), \(3\), \(4\), \(5\), \(6\)](#)
 - [ClassifyLayout](#)
 - [ColorToBW](#)
 - [Common](#)
 - [Convert](#)
 - [Common](#)
 - [Excel](#)
 - [Html](#)
 - [Images](#)
 - [Outlook](#)
 - [Pdf](#)
 - [PdfFRE](#)
 - [Rtf](#)
 - [Tiff](#)
 - [Txt](#)
 - [Word](#)
 - [Zip](#)
 - [DatacapBOX \(1\), \(2\), \(3\)](#)
 - [Dcclip](#)
 - [DCImageFix](#)
 - [DCO](#)
 - [dcpdf](#)
 - [DocumentAnalytics](#)
 - [Documentum](#)
 - [Email](#)
 - [Equalize](#)
 - [Ewsmail](#)
 - [Excel](#)
 - [Export](#)
 - [ExportDB](#)
 - [ExportXML](#)
 - [FileIO](#)
 - [FileNet P8](#)
 - [FileNetIDM](#)
 - [FingerprintMaintenance](#)
 - [FPXML](#)
 - [Grayscale](#)
 - [Html](#)
 - [IBMCM](#)
 - [ICR_C](#)

- ICR_P
- ImageConvert
- ImageFix
- Images
- Imail
- Imprint
- Intellocate
- Invoice
- IOverlay
- Locate
- Rejestrowanie
- Lookup
- Maintenance Manager
 - Aplikacja, konfiguracja
 - Przetwarzanie wsadowe
 - Rejestrowanie
 - Zapytanie, konfiguracja
 - Raportowanie
- MC_Identify
- MC_Validation
- mvscan
- OCR_A
- OCR_J
- OCR_N
- OCR_S
- OCR_SR
- OpenTextFaxServer
- Outlook
- przegląd
- PatternMatch
- Pdf
- PdfFRE
- Picture
- POLR
- Zapytanie, konfiguracja
- Recog_Shared
- Raportowanie
- rrunner
- Rtf
- sprawdzanie poprawności podpisu
- SPExport
- Podział
- Statystyki
- Tiff
- TifMerge
- TM524
- Txt
- Sprawdzanie poprawności
- Głosowanie
- Vscan
- Web Services
- Word
- Zip

- [Zones](#)
- [GoAboveWord](#), działanie (1), (2)
- [GoBelowWord](#), działanie (1), (2)
- [GoDownLine](#), działanie (1), (2)
- [GoFirstLine](#), działanie (1), (2)
- [GoFirstWord](#), działanie (1), (2)
- [GoLastLine](#), działanie (1), (2)
- [GoLastWord](#), działanie (1), (2)
- [GoLeftWord](#), działanie (1), (2)
- [GoRightWord](#), działanie (1), (2)
- [GoSiblingBlockNext](#), działanie
- [GoSiblingBlockPrevious](#), działanie
- [GoToNextFunction](#), działanie (1), (2)
- [GoUpLine](#), działanie (1), (2)
- Grayscale, działania
 - [ConvertGraytoBW](#) (1), (2)
- łączna liczba siatek elementów
 - sprawdzanie poprawności (1), (2), (3)
- Uwierzytelnianie użytkowników i grup
 - [TMA, uwierzytelnianie](#)
- Uwierzytelnianie grupy
 - [ADSI, uwierzytelnianie](#)
 - [ADSI, uwierzytelnianie grup](#)
 - Filtrowanie zadań wsadowych według grupy
 - [Monitor zadań](#)
 - [LDAP, uwierzytelnianie](#)
 - [LDAP, uwierzytelnianie grup](#)
 - [LLLDAP, uwierzytelnianie](#)
 - [Uwierzytelnianie, szablon ścieżki](#)
 - [LLLDAP, uwierzytelnianie grup](#)
- Grupa, lista uprawnień
 - GET, metoda
 - [GetGroupPermissionList](#)
- [GroupWords](#), działanie (1), (2)
- [GroupWordsLEFT](#), działanie (1), (2)
- [GroupWordsRIGHT](#), działanie (1), (2)

H

- obsługa niepowodzeń związanych z dokumentami
 - aktualizowanie aplikacji TravelDocs
 - [konfigurowanie Rulerunner przed uruchomieniem CreateDocs](#)
 - [tworzenie czynności CreateDocs](#)
 - [przenoszenie zadania tworzenia dokumentu i sprawdzanie integralności](#)
 - [uruchamianie zadania wsadowego w przepływie pracy](#)
- obsługa niepowodzeń związanych z integralnością dokumentów
 - [aktualizowanie aplikacji TravelDocs](#)
- [obsługiwanie siatek elementów w wierszach](#)
- Wydruk dokumentu, skanowanie lokalnie
 - [Datacap Desktop](#)
 - [Datacap Web Client](#)
- wymagania sprzętowe
 - [oprogramowanie, raporty o kompatybilności produktu](#)
- [HasChildOfType](#), działanie

- opis
- hr_locale
 - przegląd
- hr_locale, zmienna
 - opis
 - ustawienia języka
 - wartości nadpisywania
 - języki rozpoznania
 - ustawianie wartości
- Html, działania
 - HtmlLayout
 - HtmlPrintQuality (1), (2)
 - HtmlTiffCompression (1), (2)
 - HtmlToImage (1), (2)
- HtmlPrintQuality, działanie (1), (2)
- HtmlTiffCompression, działanie
- HtmlTiffCompression, działanie
- HtmlToImage, działanie (1), (2), (3)

I

- IBM Case Manager
 - konfigurowanie przechwytywania transakcji
 - skanowanie, przepływ pracy
 - przechwytywanie transakcji (1), (2)
- IBM Content Manager
 - Uwierzytelnianie
- IBM Content Manager Connector
 - przesyłanie plików, przykłady
- IBM Content Manager Connector, działania
 - konfigurowanie
 - przegląd
 - ustawienia parametrów
 - wymagania wstępne
- IBM Content Navigator
 - dodawanie dokumentów do zadań wsadowych
 - konfigurowanie, przechwytywanie transakcji (1), (2)
 - integrowanie
 - skanowanie, przepływ pracy
 - przechwytywanie transakcji (1), (2)
- IBM Daeja ViewONE Virtual
 - aktualizowanie
- IBM Datacap Advanced Handwriting Recognition
 - instalowanie
- dział wsparcia IBM dla oprogramowania
 - kontakt
- dział wsparcia IBM
 - rozwiązywanie problemów
 - subskrypcja aktualizacji
- IBM Support Assistant (ISA)
- IBM System Dashboard for Enterprise Content Management
- IBMCM, działania
 - IBMCM_AddPages
 - IBMCM_CreateChildItem (1), (2)

- IBMCM_CreateFolder (1), (2)
- IBMCM_CreateItem (1), (2)
- [IBMCM_DeletePages](#)
- IBMCM_Logon (1), (2)
- [IBMCM_ReplacePage](#)
- [IBMCM_SearchItem](#)
- IBMCM_SetAttributeValue (1), (2)
- IBMCM_SetChildAttributeValue (1), (2)
- IBMCM_SetDestinationFolder (1), (2)
- [IBMCM_SetFolderAttribute](#)
- [IBMCM_SetMimeType](#)
- [IBMCM_Store Folder ID In DCO](#)
- [IBMCM_Store Item In DCO](#)
- [IBMCM_StoreItemIDinDCO](#)
- IBMCM_UploadDCO_DOC (1), (2)
- IBMCM_UploadDCO_Page (1), (2)
- [IBMCM_AddPages, działanie](#)
- IBMCM_CreateChildItem, działanie (1), (2)
- IBMCM_CreateFolder, działanie (1), (2)
- IBMCM_CreateItem, działanie (1), (2)
- [IBMCM_DeletePages, działanie](#)
- IBMCM_Logon, działanie (1), (2)
- [IBMCM_ReplacePage, działanie](#)
- [IBMCM_SearchItem, działanie](#)
- IBMCM_SetAttributeValue, działanie (1), (2)
- IBMCM_SetChildAttributeValue, działanie (1), (2)
- IBMCM_SetDestinationFolder, działanie (1), (2)
- [IBMCM_SetFolderAttribute, działanie](#)
- [IBMCM_SetMimeType, działanie](#)
- [IBMCM_Store Folder ID In DCO, działanie](#)
- [IBMCM_Store Item In DCO, działanie](#)
- [IBMCM_StoreItemIDinDCO action](#)
- IBMCM_UploadDCO_DOC, działanie (1), (2)
- IBMCM_UploadDCO_Page, działanie (1), (2)
- ICR_C, działania
 - EnableLoggingICR_C (1), (2)
 - RecognizeFieldICR_C (1), (2)
 - [RecognizeFieldICR_CEx](#)
 - RecognizeFieldVoteICR_C (1), (2)
 - [RecognizePageFields ICR_CEx](#)
 - [RecognizePageFields2 CCO_ICR_C](#)
 - [RecognizePageFields2CCO_ICR_C](#)
 - RecognizePageFieldsICR_C (1), (2)
 - RecognizePageICR_C (1), (2)
 - RecognizePageToPDFICR_C (1), (2)
- ICR_P, działania
 - AddWord (1), (2)
 - DeleteWord (1), (2)
 - ImportCSF (1), (2)
 - LoadFromFile (1), (2)
 - NewDictionary (1), (2)
 - [RecognizeFieldsICR_P](#)
 - [RecognizePageFieldsICR_P](#)

- SaveToFile (1), (2)
 - SetPostalDBPathICR_P (1), (2)
- ID
 - DCO, właściwości
- Identyfikowanie, działanie (1), (2)
- identyfikowanie stron ręcznie
 - aktualizowanie aplikacji TravelDocs
 - dodawanie funkcji
 - dodawanie rozgałęzienia warunkowego do czynności ID strony
 - konfigurowanie rozgałęzień
 - konfigurowanie zestawu reguł przekazywania
 - tworzenie zadania i czynności ManualPageID
 - wykrywanie danych na niezidentyfikowanej stronie
 - uruchamianie zadania wsadowego w przepływie pracy
 - aktualizowanie zestawu reguł strony rozpoznawania
- IdentifyByBarcodesBP, działanie
- Statusy pól zignorowanych
- IIS
 - restartowanie (1), (2)
- aplikacja IIS, przetwarzanie puli
 - przetwarzanie wsadowe
 - Datacap Web Client
- IIS, aplikacja, ustawienia pul
 - dodawanie strony WWW wTM
 - Datacap Web Services
- Opcje hostingu IIS
 - Datacap Web Services
- iloc_AdjustZones, działanie (1), (2)
- iloc_AssignPageType, działanie (1), (2)
- iloc_SetDetailSimple, działanie (1), (2)
- iloc_SetDetailZones, działanie (1), (2)
- iloc_SetZones, działanie (1), (2)
- im_abort_time, działanie (1), (2)
- im_AcceptMixedAttachments, działanie
- im_AcceptNoAttachments, działanie (1), (2)
- im_done_folder, działanie (1), (2)
- im_login, działanie (1), (2)
- im_logout, działanie (1), (2)
- im_max_docs, działanie (1), (2)
- im_problem_folder, działanie (1), (2)
- skanowanie_im, działanie (1), (2)
- import_SetProxy, działanie
- import_SortByDate, działanie
- import_StoreByDate, działanie
- im_StoreEML, działanie (1), (2)
- im_types, działanie (1), (2)
- im_UseSSL, działanie (1), (2)
- czas_oczekiwania_im, działanie (1), (2)
- obraz, dostosowywanie
 - dopasowywanie wzorca
- obraz, ulepszanie
 - czyszczenie obrazów (1), (2)
 - eliminacja zakłóceń

- uruchamianie w aplikacji wzorca uczenia
 - uruchamianie we wzorcu formularzy
 - przykładowe odciski
 - kiedy zakończyć
- obraz, eksporty
 - potwierdzanie
- Image_Offset
- obraz, przetwarzanie
 - określanie ustawień
 - ustawienia
- ImageConvert, działania
 - AppendAllImages
 - AppendAllImages_ByType
 - AppendImage
 - AppendImage_StartAsNew
 - ConvertToJPEG
 - ConvertToTIFF
 - SetChrominanceFactor
 - SetDeleteOriginal
 - SetGrayScale
 - SetLuminanceFactor
 - SetTIFFCompression
- ImageDefaultDPI, działanie (1), (2)
- ImageEnhance, działanie (1), (2)
- IMAGEFILE
- ImageFileTypesToConvert, działanie (1), (2)
- ImageFix, działania
- ImageMonoThreshold, działanie
- ImageMonoThreshold, działanie
- ImageMonoType, działanie (1), (2)
- ImageName
 - DCO, właściwości
- obrazy
 - potwierdzanie eksportu
- Obrazy, działania
 - ImageDefaultDPI (1), (2)
 - ImageFileTypesToConvert (1), (2)
 - ImageMonoThreshold (1), (2)
 - ImageMonoType (1), (2)
 - ImageToTIFF (1), (2)
- ImageToTIFF, działanie (1), (2)
- Imail, działania
 - czas_przerwania_importu (1), (2)
 - import_AcceptMixedAttachments (1), (2)
 - import_AcceptNoAttachments (1), (2)
 - gotowy_folder_zaimportowany (1), (2)
 - logowanie_do_importu (1), (2)
 - wylogowywanie_z_importu (1), (2)
 - im_max_docs (1), (2)
 - folder_probleków_importu (1), (2)
 - skan_importu (1), (2)
 - import_SetProxy
 - import_SortByDate

- [import_StoreByDate](#)
 - [import_StoreEML \(1\), \(2\)](#)
 - [typy_importu \(1\), \(2\)](#)
 - [import_UseSSL \(1\), \(2\)](#)
 - [czas_oczekiwania_na_import \(1\), \(2\)](#)
- [ImgEnter, klient WWW](#)
- [importuj](#)
 - [klucze szyfrowania \(1\), \(2\), \(3\), \(4\), \(5\), \(6\), \(7\), \(8\)](#)
 - [rozwiązywanie problemów](#)
- [Import](#)
- [ImportAsDocumentType](#)
- [ImportAsDocumentType, działanie](#)
- [ImportCSF, działanie \(1\), \(2\)](#)
- [ImportFaxes, działanie \(1\), \(2\)](#)
- [ImportLimit, działanie](#)
- [Imprint, działanie](#)
- [ImPrint, działanie](#)
- [Imprint, działania](#)
 - [AnnotateImage \(1\), \(2\)](#)
 - [Imprint](#)
 - [ImPrint](#)
 - [Redact \(1\), \(2\)](#)
 - [RedactByRegEx \(1\), \(2\)](#)
 - [RedactParameters \(1\), \(2\)](#)
 - [SetAdjustedWidth \(1\), \(2\)](#)
 - [SetFontName \(1\), \(2\)](#)
 - [SetFontSize \(1\), \(2\)](#)
 - [SetOpaque \(1\), \(2\)](#)
- [IncrementBatchVar, działanie \(1\), \(2\)](#)
- [Indeks](#)
- [indeks, dane](#)
 - [przechwytywanie w FastDoc](#)
 - [potwierdzanie eksportu](#)
 - [korzystanie z klucza klikania N](#)
- [indeks, eksporty danych](#)
 - [potwierdzanie](#)
- [indeks, uzupełniania pól](#)
 - [definiowanie za pomocą bazy danych](#)
- [indeks, sprawdzanie poprawności pól](#)
 - [definiowanie za pomocą bazy danych](#)
- [indeks, pola](#)
 - [konfigurowanie zestawu reguł w FastDoc](#)
 - [definiowanie](#)
 - [korzystając ze słów kluczowych](#)
- [IndexProperty_ID_Component action \(1\), \(2\)](#)
- [IndexProperty_ID_DateComponent, działanie](#)
- [IndexProperty_ID_Value, działanie](#)
- [IndexProperty_LeftJUSTIFY, działanie \(1\), \(2\)](#)
- [IndexProperty_RightJUSTIFY, działanie \(1\), \(2\)](#)
- [IndexProperty_SmartParameter, działanie \(1\), \(2\)](#)
- [IndexProperty_ID_Value, działanie](#)
- [poszczególne pozycje pól](#)
 - [dostosowywanie do dopasowania wzorca](#)

- [InheritParentPosition](#), działanie (1), (2)
- [InheritSnippets](#), działanie
- [InitializeEngine](#), działanie (1), (2)
- [InsertChars](#), działanie (1), (2)
- [InsertDecimalPoint](#), działanie (1), (2)
- [Inst_Verify](#), czynność
- instalowanie
 - [często wykorzystywane parametry](#)
 - [Datacap Maintenance Manager](#)
 - [uruchamianie z wiersza komend](#)
- Instalowanie
 - Lista kontrolna
 - [Datacap, serwer](#)
 - [Datacap, komponenty](#)
- instalowanie i konfigurowanie
 - [na jednym komputerze](#)
- Instalowanie i konfigurowanie
 - [Zastosowanie licencji](#)
 - Lista kontrolna
 - [Datacap, serwer](#)
 - [Na jednym komputerze](#)
- instalacja
 - [parametry wiersza komend](#)
 - [Datacap, serwer](#) (1), (2)
 - [Datacap Web Client](#)
 - [Datacap Web Client](#), przesyłanie, usługa (1), (2)
 - [FastDoc](#)
 - [Fingerprint](#), usługa (1), (2)
 - [Maintenance Manager](#)
 - [Microsoft Windows Installer](#), parametry
 - [wymagania wstępne](#) (1), (2)
 - [Report Viewer](#) (1), (2), (3), (4)
 - [przegląd](#) (1), (2)
 - [wymagane czynności](#) (1), (2)
 - [Rulerunner](#) (1), (2), (3)
 - [uruchamianie z wiersza komend](#) (1), (2)
 - [konfigurowanie Report Viewer](#)
 - [komponenty innych firm w Rulerunner](#)
 - weryfikowanie
 - [Connector, działania](#)
- Instalacja i konfiguracja
 - Lista kontrolna
 - [Rulerunner](#)
- Parametry instalacji
 - [Program instalacyjny](#)
- planowanie instalacji
 - [Datacap](#)
 - [FastDoc](#), wymagania wstępne
 - [wymagania wstępne](#)
 - [wybieranie języka systemu Windows dla FastDoc](#)
- Program instalacyjny
 - [Parametry instalacji](#)
- [Parametry instalacji programu](#)

- Kroki instalacji
 - [Datacap Web Services](#)
- installationRulerunner (1), (2)
 - [Datacap Server, uprawnienia](#)
 - [zamykanie oprogramowania Datacap](#)
- instalowanie
 - [Datacap Navigator \(1\), \(2\), \(3\), \(4\)](#)
- Instalowanie
 - [Datacap, klient](#)
 - [Datacap Web Client](#)
 - [Datacap Web Services](#)
- Instalowanie i konfigurowanie
 - [Datacap Web Client](#)
- Instalowanie i konfigurowanie Datacap
 - [Na jednym komputerze](#)
- instalowanie komponentów
 - [na jednym komputerze](#)
- integrowanie
- Intellocate, działania
 - [iloc_AdjustZones \(1\), \(2\)](#)
 - [iloc_AssignPageType \(1\), \(2\)](#)
 - [iloc_SetDetailZones \(1\), \(2\)](#)
 - [iloc_SetZones \(1\), \(2\)](#)
 - [IsPageDataMissing \(1\), \(2\)](#)
- interfejs, sprawdzanie poprawności
 - [DCO, metody](#)
 - [IsValid](#)
- Internet Information Services
 - [restartowanie \(1\), \(2\)](#)
- Invoice, działania
 - [AddToDetailErrorMsg \(1\), \(2\)](#)
 - [AddToErrorMsg \(1\), \(2\)](#)
 - [AllMixedCase \(1\), \(2\)](#)
 - [AllowOnlyChars \(1\), \(2\)](#)
 - [AlterDatebyDay \(1\), \(2\)](#)
 - [CalculateNotesZone \(1\), \(2\)](#)
 - [CaptureOpInfo \(1\), \(2\)](#)
 - [CheckAndFixDecimal \(1\), \(2\)](#)
 - [CheckForSticky \(1\), \(2\)](#)
 - [CheckFreeDiskSpace \(1\), \(2\)](#)
 - [ClearErrorMsg \(1\), \(2\)](#)
 - [CreateFingerprint \(1\), \(2\)](#)
 - [DetailFix \(1\), \(2\)](#)
 - [DoMsgbox \(1\), \(2\)](#)
 - [ExecuteSQLBind \(1\), \(2\)](#)
 - [FindExportImage \(1\), \(2\)](#)
 - [FPXMLUsed \(1\), \(2\)](#)
 - [GenerateDetails \(1\), \(2\)](#)
 - [iloc_SetDetailSimple \(1\), \(2\)](#)
 - [IncrementBatchVar \(1\), \(2\)](#)
 - [Is_InCharSet \(1\), \(2\)](#)
 - [Is_JobName \(1\), \(2\)](#)
 - [Is_JobNamePrefix](#)

- IsChildFieldBlank (1), (2)
- IsChildFieldValue (1), (2)
- IsCurrentObjValue (1), (2)
- IsCurrentObjVariable (1), (2)
- IsFingerPrintClass (1), (2)
- IsInINI (1), (2)
- IsInList (1), (2)
- IsMultipageDocument (1), (2)
- IsSinglePageDocument (1), (2)
- IsStationIDSuffix (1), (2)
- IsTaskName (1), (2)
- JobNamePrefix
- LoadCCOFromField (1), (2)
- MovePDF (1), (2)
- OpenConnection (1), (2)
- ParseImageName (1), (2)
- PopulateZNLineItem FieldDynamic (1), (2)
- ReadFPXMLZones (1), (2)
- SaveObjectVariable (1), (2)
- ScanLineItemDynamic (1), (2)
- SendOutlookNotification (1), (2)
- SetDynamicDetailZones (1), (2)
- SetPicChar (1), (2)
- SetStickyNo (1), (2)
- SetToDocIDMPTIFF (1), (2)
- SwapImages (1), (2)
- SwitchMMDD (1), (2)
- UpdateFPStats (1), (2)
- ValidateVendor (1), (2)
- WriteErrorMessage (1), (2)
- faktura, obrazy
 - powiązanie dostawcy z odciskiem
 - przechwytywanie pól na obróconym obrazie
 - czyszczenie (1), (2)
 - eksportowanie (1), (2)
 - obsługa nieznanymi faktur
 - identyfikowanie wierszy szczegółów na nieznanymi fakturach
 - logowanie do Datacap Accounts Payable na kliencie Datacap Web Client
 - przygotowanie do przetwarzania
 - przetwarzanie (1), (2), (3), (4), (5), (6)
 - przetwarzanie wielu nieznanymi faktur
 - rozwiązywanie problemów związanych z nieznanymi dostawcami
 - uruchamianie Datacap Accounts Payable
 - uruchamianie Datacap Accounts Payable na kliencie Datacap Web Client
 - uruchomienie czynności Batch Profiler (1), (2)
 - uruchomienie czynności Eksportu (1), (2)
 - uruchomienie czynności FlexID
 - uruchomienie czynności skanowania
 - na Datacap Web Client
 - uruchomienie czynności skanowania używając Datacap Desktop
 - uruchomienie czynności przesyłania
 - na Datacap Web Client
 - uruchomienie czynności Weryfikacji

- na Datacap Desktop
 - na Datacap Web Client
 - skanowanie
 - na Datacap Web Client
 - skanowanie za pomocą Datacap Desktop
 - konfigurowanie dla przetwarzania
 - dodawanie elementów wiersza zamówienia
 - dodawanie dostawców do listy Demo Vendors
 - skróty czynności (1), (2)
 - przesyłanie
 - na Datacap Web Client
 - Weryfikowanie instrukcji w oknie
 - weryfikowanie danych na fakturze
 - na Datacap Desktop
 - na Datacap Web Client
- iOS
 - automatyczny strefowy współczynnik OCR
 - klasyfikacja kodów kreskowych
 - geokodowanie i lokalizacja
- IOverlay, działania
 - Overlay (1), (2)
 - setBackgroundImage (1), (2)
 - SetDitheringBackground (1), (2)
 - SetHaloBackground (1), (2)
- Is_InCharSet, działanie (1), (2)
- Is_JobName, działanie (1), (2)
- Is_JobNamePrefix, działanie
- IsAlpha, działanie (1), (2)
- IsBatchAborted, działanie
- IsBlankPage, działanie (1), (2)
- IsChildFieldBlank, działanie (1), (2)
- IsChildFieldValue, działanie (1), (2)
- IsCurrency, działanie (1), (2)
- IsCurrentObjValue, działanie (1), (2)
- IsCurrentObjVariable, działanie (1), (2)
- IsDate_FormatEuro, działanie
 - opis
- IsDateValue, działanie (1), (2)
- IsDirectoryPresent, działanie (1), (2)
- IsDocumentCountMoreThan, działanie
- IsError
 - DCO, metody
- IsFieldCurrency, działanie (1), (2)
- IsFieldDate, działanie (1), (2)
- IsFieldDateEqualOrAfter, działanie (1), (2)
- IsFieldDateEqualOrBefore, działanie (1), (2)
- IsFieldDateUpToToday, działanie (1), (2)
- IsFieldDateWithinRange, działanie (1), (2)
- IsFieldDateWithinXDays, działanie (1), (2)
- IsFieldDateWithReformat, działanie (1), (2)
- IsFieldEmpty, działanie (1), (2)
- IsFieldFilled, działanie (1), (2)
- IsFieldGreaterOrEqual, działanie (1), (2)

- IsFieldHidden, działanie (1), (2)
- IsFieldLengthMax, działanie (1), (2)
- IsFieldLengthMin, działanie (1), (2)
- IsFieldLessOrEqual, działanie (1), (2)
- IsFieldLocalCurrency, działanie
 - opis
- IsFieldMatching, działanie (1), (2)
- IsFieldPercentAlpha, działanie (1), (2)
- IsFieldPercentNonNumeric, działanie (1), (2)
- IsFieldPercentNumeric, działanie (1), (2)
- IsFilePresent, działanie (1), (2)
- IsFileReadOnly, działanie (1), (2)
- IsFingerPrintClass, działanie (1), (2)
- IsFirstDocInBatch, działanie
 - opis
- [IsFirstDocumentInBatch](#), działanie
- IsInINI, działanie (1), (2)
- IsInList, działanie (1), (2)
- IsInvoiceFromUS, działanie
 - opis
- ISIS, skaner
 - [Ustawienie separatorów w dokumencie](#)
- ISIS, konfigurowanie skanera
 - [FastDoc](#)
- IsLocalDecimalSeparator, działanie
 - opis
- IsMatchingJobID, działanie (1), (2)
- IsMaxOMRChecked, działanie (1), (2)
- IsMinOMRChecked, działanie (1), (2)
- IsMultipageDocument, działanie (1), (2)
- IsNumber, działanie (1), (2)
- IsOriginalEuroFormat, działanie
 - opis
- IsPageDataMissing, działanie (1), (2)
- IsPatternInField, działanie (1), (2)
- IsProfilePresent, działanie (1), (2)
- IsRoute
 - [DCO, metody](#)
- [IsSelectedBlockType](#), działanie
- IsSinglePageDocument, działanie (1), (2)
- IsStationIDSuffix, działanie (1), (2)
- IsSupportedImageFile, działanie (1), (2)
- IsTaskName, działanie (1), (2)
- IsThisFieldEmpty, działanie (1), (2)
- IsThisFieldFilled, działanie (1), (2)
- IsValid
 - [DCO, metody](#)
- IsValue, działanie (1), (2)
- IsValue_RegEx, działanie (1), (2)
- IsVariableEmpty, działanie (1), (2)
- IsVariableFilled, działanie (1), (2)
- IsWorkstationLocale, działanie
 - opis

- Włoski
 - [kody języków](#)
- elementy
 - [grupowanie](#)

J

- działanie, informacje
 - [uzyskiwanie dostępu \(1\), \(2\)](#)
- [Monitor zadań](#)
 - [Tworzenie kolumny niestandardowej](#)
 - [Datacap Navigator \(1\), \(2\)](#)
 - [Datacap Web Client](#)
 - [konfigurowanie](#)
- [monitorowanie zadań](#)
- [JobNamePrefix, działanie](#)
- zadania
 - [konfigurowanie w FastDoc \(1\), \(2\)](#)
 - [tworzenie](#)
 - [tworzeniu w celu obsługi warunków specjalnych](#)
 - [opis](#)
- [JoinPreviousDocument, działanie](#)

K

- nazwa kluczowa
 - [określanie dla parametrów inteligentnych](#)
- skróty klawiaturowe
 - [dodawanie](#)
 - [aspx, strony WWW](#)
 - [Datacap Web Client](#)
 - [FastDoc](#)
- Skróty klawiszowe
 - [Datacap Desktop \(1\), \(2\), \(3\)](#)
 - [Poprawka, czynność](#)
 - [Skan, czynność](#)
 - [Weryfikowanie, czynność](#)
- słowo kluczowe, listy
 - [korzystanie za pomocą dopasowywania tekstu](#)
- bazy wiedzy
 - [wyszukiwanie rozwiązań problemów](#)

L

- Etykieta
 - [Konfiguracja DCO](#)
 - [Weryfikacja, panel](#)
 - [Datacap Desktop](#)
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
- kody języków
 - [Chiński](#)
 - [Holenderski](#)
 - [Wschodnioeuropejski](#)
 - [Angielski](#)

- Francuski
 - Niemiecki
 - Włoski
 - Portugalski
 - Rosyjski
 - ustawienie
 - Hiszpański
 - Szwedzki
- język, rozpoznanie
 - ustawienia języka
- ustawienia języka
 - ustawienia językowe
 - hr_locale, zmienna
 - rozpoznanie
- LAST_RR_PROFILE
- Szerokość geograficzna
- układ, pliki XML
 - tworzenie paneli Datacap Desktop (1), (2)
 - generowanie (1), (2)
- układ, przegląd plików XML (1), (2)
- LDAP, uwierzytelnianie
 - włączanie dla Report Viewer
 - Uwierzytelnianie grupy
- Learn_Zones, działanie
 - opis
- Learn_ZonesFPX, działanie
 - opis
- Uczenie, wzorzec
 - czyszczenie obrazów
 - konfigurowanie w FastDoc
 - zadania
 - lokalizowanie pól
 - przetwarzanie dokumentów
 - konfigurowanie sprawdzania poprawności pól
 - konfigurowanie pól w FastDoc
- LeftTruncate, działanie (1), (2)
- Library_DMA_Initialize, działanie (1), (2)
- Library_DS_Initialize, działanie (1), (2)
- Library_IS_Initialize, działanie (1), (2)
- Library_LogIn, działanie
- Library_LogIn, działanie
- Library_LogOff, działanie
- Library_LogOff, działanie
- ograniczenia
 - dopasowywanie tekstu w celu rozpoznawania danych
- wiersz, dane siatki elementów
 - dołączanie reguł do hierarchii dokumentu
 - eksportowanie
 - eksportowanie do pliku XML
 - dodawanie reguł do zestawu danych ExportXML
 - dołączanie reguł Export Other XML do hierarchii dokumentu
 - uruchamianie zadania wsadowego w przepływie pracy
- wiersz, strony siatki elementów

- weryfikowanie
- siatki elementów w wierszach
 - dodawanie istniejących reguł stron
 - dołączanie reguł do hierarchii dokumentu
 - dołączanie reguł sprawdzania poprawności do DCO
 - tworzenie pól danych
 - tworzenie odcisków na stronie
 - tworzenie reguł sprawdzania poprawności
 - definiowanie hierarchii dokumentu
 - definiowanie, strefy rozpoznawania
 - eksportowanie danych
 - lokalizowanie pól
 - rozpoznawanie, reguła
 - tworzenie w łącznej siatce elementów
 - rozpoznawanie danych
 - usuwanie elementów nieprzypisanych do wierszy
 - reguły
 - reguły dotyczące usuwanie elementów nieprzypisanych do wierszy
 - uruchamianie zadania wsadowego w przepływie pracy
 - aktualizowanie hierarchii dokumentu
 - sprawdzanie poprawności danych
 - sprawdzanie poprawności łącznej liczby elementów w wierszu
- ListItem_AddElement, działanie (1), (2)
- ListItem_BlankFields, działanie (1), (2)
- ListItem_ClearElements, działanie (1), (2)
- ListItem_ExportElements action (1), (2)
- ListItem_SmartParameter, działanie (1), (2)
- LLDAP, uwierzytelnianie
 - Uwierzytelnianie użytkowników
- LLDAP, uwierzytelnianie grup
 - Uwierzytelnianie, wzorzec ścieżki
 - Konfiguracja niestandardowych katalogów
- Równoważenie obciążenia
 - Datacap, serwery
 - Datacap Report Viewer, serwery
 - Datacap Web Client, serwery
 - Datacap Web Services, serwery
 - Fingerprint, usługa, serwery
- LoadBlockCCO, działanie (1), (2)
- LoadCCOFromField, działanie (1), (2)
- LoadFromFile, działanie
- LoadFromFile, działanie
- LoadSettings, działanie
- LoadSettings, działanie
- LoadSettings_FingerprintID, działanie
- LoadSettings_FingerprintID, działanie
- LoadZones, działanie (1), (2)
- Tryb lokalny
 - FastDoc
 - konfigurowanie aplikacji
 - konfigurowanie profili zadań wsadowych
 - runningFastDoc
 - uruchamianie FastDoc

- [Lokalny, konfigurowanie skanera](#)
- lokalne, konto w systemie
 - [dodawanie do grupy administracyjnej](#)
- zmienne ustawień narodowych
 - [hr_locale](#)
 - [dziedziczenie, reguły](#)
 - [nadpisywanie](#)
 - [ustawienie \(1\), \(2\)](#)
- Lokalizacja, działania
 - [AddKeyList \(1\), \(2\)](#)
 - [AggregateKeyList \(1\), \(2\)](#)
 - [DefaultValue \(1\), \(2\)](#)
 - [FilterIt \(1\), \(2\)](#)
 - [FindDBList \(1\), \(2\)](#)
 - [FindDBList_InZone \(1\), \(2\)](#)
 - [FindKeyList \(1\), \(2\)](#)
 - [FindKeyList_InZone \(1\), \(2\)](#)
 - [FindLastKeyList \(1\), \(2\)](#)
 - [FindLastKeyList_InZone \(1\), \(2\)](#)
 - [FindLastRegEx \(1\), \(2\)](#)
 - [FindLastRegEx_InZone \(1\), \(2\)](#)
 - [FindLastRegExList \(1\), \(2\)](#)
 - [FindLastRegExList_InZone \(1\), \(2\)](#)
 - [FindLastWord \(1\), \(2\)](#)
 - [FindLastWord_InZone \(1\), \(2\)](#)
 - [FindNextDBList \(1\), \(2\)](#)
 - [FindNextDBList_InZone \(1\), \(2\)](#)
 - [FindNextKeyList \(1\), \(2\)](#)
 - [FindNextKeyList_InZone \(1\), \(2\)](#)
 - [FindNextRegExList \(1\), \(2\)](#)
 - [FindNextRegExList_InZone](#)
 - [FindNextRegExList_InZone](#)
 - [FindRegExList \(1\), \(2\)](#)
 - [FindRegExList_InZone \(1\), \(2\)](#)
 - [GetSelectedBlockType](#)
 - [GoAboveWord \(1\), \(2\)](#)
 - [GoBelowWord \(1\), \(2\)](#)
 - [GoDownLine \(1\), \(2\)](#)
 - [GoFirstLine \(1\), \(2\)](#)
 - [GoFirstWord \(1\), \(2\)](#)
 - [GoLastLine \(1\), \(2\)](#)
 - [GoLastWord \(1\), \(2\)](#)
 - [GoLeftWord \(1\), \(2\)](#)
 - [GoRightWord \(1\), \(2\)](#)
 - [GoSiblingBlockNext](#)
 - [GoSiblingBlockPrevious](#)
 - [GoUpLine \(1\), \(2\)](#)
 - [GroupWords \(1\), \(2\)](#)
 - [GroupWordsLEFT \(1\), \(2\)](#)
 - [GroupWordsRIGHT \(1\), \(2\)](#)
 - [IsAlpha \(1\), \(2\)](#)
 - [IsCurrency \(1\), \(2\)](#)
 - [IsDateValue \(1\), \(2\)](#)

- [IsNumber \(1\), \(2\)](#)
- [IsSelectedBlockType](#)
- [IsValue \(1\), \(2\)](#)
- [IsValue_RegEx \(1\), \(2\)](#)
- [LocatePositionRestore](#)
- [LocatePositionSave](#)
- [MaxLength \(1\), \(2\)](#)
- [MergeWordLF \(1\), \(2\)](#)
- [MergeWordRT \(1\), \(2\)](#)
- [MinLength \(1\), \(2\)](#)
- [RegExFind \(1\), \(2\)](#)
- [RegExFind_InBlock](#)
- [RegExFind_InZone \(1\), \(2\)](#)
- [RegExFindNext \(1\), \(2\)](#)
- [RegExFindNext_InZone](#)
- [RegExFindNext_InBlock](#)
- [ScanRT \(1\), \(2\)](#)
- [SelectParentBlock](#)
- [SelectParentBlockOuterType](#)
- [SelectParentBlockType](#)
- [SelectSnippet \(1\), \(2\)](#)
- [SetRect \(1\), \(2\)](#)
- [UpdateDCOField \(1\), \(2\)](#)
- [UpdateField \(1\), \(2\)](#)
- [UpdateFieldWithBlock](#)
- [ValueInField \(1\), \(2\)](#)
- [ValueInField_Fuzzy \(1\), \(2\)](#)
- [ValueInField_RegEx \(1\), \(2\)](#)
- [WordFind \(1\), \(2\)](#)
- [WordFind_InZone \(1\), \(2\)](#)
- [WordFind_Offset \(1\), \(2\)](#)
- [WordFindNext \(1\), \(2\)](#)
- [WordFindNext_InZone \(1\), \(2\)](#)
- [LocatePositionRestore, działanie](#)
- [LocatePositionSave, działanie](#)
- lokalizowanie, pola
 - [konfigurowanie we wzorcu uczenia, aplikacje](#)
- [Lokalizacja](#)
- [dziennik, pliki \(1\), \(2\)](#)
 - [Connector, działania](#)
 - [testowanie w Datacap Studio](#)
 - [RRS](#)
 - [Rulerunner](#)
 - [czynności](#)
- [Wylogowywanie z aplikacji](#)
 - [POST, metoda](#)
 - [Wylogowanie](#)
- [Logowanie jako przywilej usługi](#)
 - [Datacap Server, właściwości](#)
 - [Rulerunner, usługa, właściwości](#)
- [logowanie jako usługa](#)
 - [nadawanie uprawnień do Rulerunner](#)
- [Logowanie do aplikacji](#)

- POST, metoda
 - [Zalogowanie się](#)
- [LogClear](#), działanie (1), (2)
- [LogConfigure](#), działanie (1), (2)
- rejestrowanie
 - [Rulerunner](#)
- Rejestrowanie, działania
 - [LogClear](#) (1), (2)
 - [LogConfigure](#) (1), (2)
 - [LogSendEmail](#) (1), (2)
 - [LogWriteEventLog](#) (1), (2)
 - [LogWriteRecordSet](#) (1), (2)
 - [LogWriteSQLQuery](#) (1), (2)
- zalogowanie
 - [Datacap Web Client](#)
 - [Report Viewer](#)
- Dzienniki
 - Rulerunner, dzienniki
 - [Ustawianie rejestrowania wg aplikacji](#)
 - [Ustawianie rejestrowania wg czynności](#)
- [LogSendEmail](#), działanie (1), (2)
- [LogWriteEventLog](#), działanie (1), (2)
- [LogWriteRecordSet](#), działanie (1), (2)
- [LogWriteSQLQuery](#), działanie (1), (2)
- [Długość geograficzna](#)
- [LookForExtensions](#)
- [LookforExtensions](#), działanie
- Wyszukiwanie
 - [Konfiguracja DCO](#)
 - Weryfikacja, panel
 - [Datacap Desktop](#)
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
- Wyszukiwanie, działania
 - [ClearLookupResults](#) (1), (2)
 - [CloseConnection](#) (1), (2)
 - [ExecuteSQL](#) (1), (2)
 - [OpenConnection](#) (1), (2)
 - [PopulateWithResult](#) (1), (2)
 - [SmartSQL](#) (1), (2)
- wyszukiwanie, baza danych
 - [potwierdzenie typu samochodu](#)
- wyszukiwanie, tabela bazy danych
 - [tworzenie](#)
- [LookupEx](#)
 - [Konfiguracja DCO](#)
 - Weryfikacja, panel
 - [Datacap Desktop](#)
 - [Datacap Web Client](#)

M

- [Maintenance Manager](#)
 - instalowanie komponentów dla programistów (1), (2)

- Maintenance Manager, działania
 - [Aplikacja, konfiguracja](#)
 - [SetAdminDB \(1\), \(2\)](#)
 - [SetApplication \(1\), \(2\)](#)
 - [SetEngineDB \(1\), \(2\)](#)
 - [SetPassword \(1\), \(2\)](#)
 - [SetServer \(1\), \(2\)](#)
 - [SetStation \(1\), \(2\)](#)
 - [SetupDisconnectAll \(1\), \(2\)](#)
 - [SetupOpenApplication \(1\), \(2\)](#)
 - [SetupOpenApplicationEx \(1\), \(2\)](#)
 - [SetUser \(1\), \(2\)](#)
 - [Przetwarzanie wsadowe](#)
 - [ProcessChange BatchStatus](#)
 - [ProcessChangeBatch StatusOrder \(1\), \(2\)](#)
 - [ProcessChangeBatch StatusTaskOrder](#)
 - [ProcessChangeBatchStatus](#)
 - [ProcessChangeBatchStatus TaskOrder](#)
 - [ProcessClearAuditTable \(1\), \(2\)](#)
 - [ProcessClearDebugTable \(1\), \(2\)](#)
 - [ProcessDeleteBatches \(1\), \(2\)](#)
 - [ProcessDeleteBatchesEx \(1\), \(2\)](#)
 - [ProcessInjectBatches \(1\), \(2\)](#)
 - [ProcessMoveBatches \(1\), \(2\)](#)
 - [ProcessMoveBatchesEx](#)
 - [ProcessMoveBatchesEX](#)
 - [ProcessMoveDBRecords \(1\), \(2\)](#)
 - [ProcessReset Pending OrNotify](#)
 - [ProcessResetPendingOrNotify](#)
 - [ProcessRunSqlQuery \(1\), \(2\)](#)
 - [ProcessRunSqlQueryEx](#)
 - [Rejestrowanie](#)
 - [LogClear \(1\), \(2\)](#)
 - [LogConfigure \(1\), \(2\)](#)
 - [LogSendEmail \(1\), \(2\)](#)
 - [LogWriteEventLog \(1\), \(2\)](#)
 - [LogWriteRecordSet \(1\), \(2\)](#)
 - [LogWriteSQLQuery \(1\), \(2\)](#)
 - [Zapytanie, konfiguracja](#)
 - [QueryClear \(1\), \(2\)](#)
 - [QuerySetAge \(1\), \(2\)](#)
 - [QuerySetBatchRange \(1\), \(2\)](#)
 - [QuerySetBranch \(1\), \(2\)](#)
 - [QuerySetDateFormat \(1\), \(2\)](#)
 - [QuerySetDateRange \(1\), \(2\)](#)
 - [QuerySetDateTimeFormat \(1\), \(2\)](#)
 - [QuerySetGeneric \(1\), \(2\)](#)
 - [QuerySetJobID \(1\), \(2\)](#)
 - [QuerySetOperator \(1\), \(2\)](#)
 - [QuerySetPriority \(1\), \(2\)](#)
 - [QuerySetSeparator \(1\), \(2\)](#)
 - [QuerySetStation \(1\), \(2\)](#)
 - [QuerySetStatus \(1\), \(2\)](#)

- QuerySetTaskID (1), (2)
 - Raportowanie
 - ReportQueryTMUsage (1), (2)
 - ReportSetReportingTable (1), (2)
 - ReportSetUsageDBTable (1), (2)
- MakeFieldHighConfidence, działanie
 - opis
- ręczna identyfikacja stron
 - dodawanie funkcji
 - dodawanie kierowania w celu odblokowania
 - dodawanie rozgałęzienia warunkowego do czynności ID strony
 - konfigurowanie rozgałęzień
 - konfigurowanie zestawu reguł przekazywania
 - tworzenie zadania i czynności ManualPageID
 - wykrywanie danych na niezidentyfikowanej stronie
 - uruchamianie zadania wsadowego w przepływie pracy
 - aktualizowanie zestawu reguł strony rozpoznawania
- Ręczna identyfikacja stron i rejestracja
- ManualIDValidate, reguła
 - tworzenie
 - testowanie
- ManualPageID
 - aktualizowanie
- ManualPageID, zadanie
 - tworzenie
- ManualPageID, ustawienie
 - edytowanie
- ManualPageID, czynność
 - tworzenie
- MatchBarCode, działanie (1), (2)
- MatchBarcodeBP, działanie (1), (2)
- MatchBarcodePrefixBP, działanie (1), (2)
- MatchPattern, działanie (1), (2)
- MAX_TYPES
- MaxLength
 - Konfiguracja DCO
 - Weryfikacja, panel
 - Datacap Web Client
- MaxLength, działanie (1), (2)
- MC_Identify, działanie
 - AutoField (1), (2)
 - FindFields (1), (2)
 - ReadDCOSetup (1), (2)
 - ReadPageSetup (1), (2)
 - SetFormType (1), (2)
 - SetMaxTolerantDistance (1), (2)
- MC_ReadZones, działanie
- MC_Validation
- MC_Validation, działanie
 - AddCenturyTo2YearDigit (1), (2)
 - AddToDetailErrorMsg
 - AddToDetailMsg
 - AddToErrorMsg (1), (2)

- CalculateHCFALineCharges (1), (2)
- CalculateUBLLineCharges (1), (2)
- CheckDocID (1), (2)
- ClearErrorMsg (1), (2)
- [CommonParseAddress](#)
- [CommonParseAddresser](#)
- CommonValAddress (1), (2)
- ConvertHyphen (1), (2)
- FilterPID (1), (2)
- FormatFieldLengths (1), (2)
- InheritSnippets (1), (2)
- MC_ReadZones (1), (2)
- Parse31aPhSig (1), (2)
- Parse58ainsnm (1), (2)
- [Parse58binsnm](#)
- Parse58cinsnm (1), (2)
- ParseConditionCodes (1), (2)
- ParseEPSDT (1), (2)
- ParseLastFirstIniNames (1), (2)
- ParseNDC (1), (2)
- PopulateFromField (1), (2)
- SetConf (1), (2)
- SetOriginalTIF (1), (2)
- StripTrailingAlpha (1), (2)
- TransformLI (1), (2)
- UpdateCredentialList (1), (2)
- ValidateNPI (1), (2)
- ValProcedureCode (1), (2)
- ValRequiredCode (1), (2)
- MCCOPositionAdjust, działanie (1), (2)
- medical claim, formularze
 - [Identyfikowanie](#)
- Medical Claims
 - [5010, konfigurowanie formularza](#)
 - [działania](#)
 - [konfigurowanie](#)
 - [Datacap Web Client](#)
 - [logowanie](#)
 - [uprawnienia](#)
 - [skanowanie](#)
 - [przesyłanie](#)
 - [weryfikowanie](#)
 - [przegląd](#)
 - [uprawnienia](#)
 - [skrót](#) (1), (2)
 - [czynności](#) (1), (2)
 - [Poprawka](#)
- Medical Claims Capture
 - [5010, konfiguracja formularza dla instytucji](#)
 - [5010, konfiguracja formularza firmowego](#)
- Medical Claims, czynności
 - [W tle](#)
 - [Skanowanie](#)

- [Weryfikowanie \(Prof_Verify i Inst_Verify\)](#)
 - [VScan](#)
- [MergeCCOs_ByType](#), działanie (1), (2)
- [MergeLineItem FieldToPageField](#), działanie
 - [opis](#)
- [MergePageFieldToDocVar](#), działanie
 - [opis](#)
- [MergeWordLF](#), działanie (1), (2)
- [MergeWordRT](#), działanie (1), (2)
- [MergeZones](#), działanie (1), (2)
- [MESSAGE](#)
- [MessageBox](#), działanie (1), (2)
- [MessageID](#), działanie
- [MessageIDParameter](#), działanie
- [METRIC](#)
- [Microsoft .Net Framework 3.5.1](#)
- [Microsoft .Net Framework 4.0](#)
- [Microsoft aspnet_regiis.exe](#)
- [Microsoft Internet Information Services](#)
 - [Fingerprint, usługa](#)
- [Microsoft SQL Server, baza danych](#)
 - [planowanie](#)
 - [wymagania wstępne](#)
 - [wymagane oprogramowanie](#)
- [Microsoft SQL Server, bazy danych](#)
 - [konfigurowanie](#)
 - [tworzenie pojedynczych użytkowników](#)
 - [planowanie](#)
 - [zapewnianie dostępu](#)
- [Microsoft Windows Installer, parametry](#)
 - [przeglądanie](#)
- [Microsoft Windows, parametry](#)
- [migrowanie](#)
 - [Application Wizard \(1\), \(2\)](#)
 - [aplikacje \(1\), \(2\), \(3\), \(4\), \(5\), \(6\), \(7\), \(8\)](#)
 - [przekształcanie paneli \(1\), \(2\), \(3\), \(4\), \(5\), \(6\), \(7\)](#)
 - [tworzenie paneli Datacap Desktop \(1\), \(2\)](#)
 - [z poprzedniego wydania](#)
 - [generowanie układu, pliki XML \(1\), \(2\)](#)
 - [przeglądanie układu, pliki XML \(1\), \(2\)](#)
- [migracja](#)
 - [aplikacje](#)
 - [bazy danych](#)
 - [przegląd](#)
- [opcje migrowania](#)
 - [aplikacje](#)
 - [bazy danych](#)
- [MIN_TYPES](#)
- [MinLength](#), działanie (1), (2)
- [mobilna](#)
 - [konfigurowanie](#)
- [aplikacja mobilna](#)
 - [automatyczny strefowy współczynnik OCR](#)

- [klasyfikacja kodów kreskowych](#)
 - [geokodowanie i lokalizacja](#)
- Mobilne, profile przechwytywania
 - GET, metoda
 - [GetMobileProfiles](#)
- [Monitorowanie systemu użytkownika za pomocą Datacap Navigator](#)
- przenoszenie
 - [aplikacje](#)
 - [bazy danych](#)
- MoveChild
 - [DCO, metody](#)
- MoveImageFileToDirectory, działanie (1), (2)
- MoveIn
 - [DCO, metody](#)
- MovePDF, działanie (1), (2)
- [Weryfikacja metodą wielokrotnego wprowadzania danych](#)
 - [ustawienia](#)
- MultiLine
 - [Konfiguracja DCO](#)
 - Weryfikacja, panel
 - [Datacap Desktop](#)
- MultiPunch
 - [Konfiguracja DCO](#)
 - Weryfikacja, panel
 - [Datacap Desktop](#)
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
- mvscan, działania
 - [folder_zachowywania_mv](#)
 - skan (1), (2)
 - [ustawienie_czasu_oczekiwania](#) (1), (2)
 - [ustawienie_kopiowania_folder](#) (1), (2)
 - [ustawienie_usunięcia_pustych_folderów](#) (1), (2)
 - [ustawienie_folderu](#) (1), (2)
 - [ustawienie_sprawdzania_poprawności_obrazu](#) (1), (2)
 - [set_max_docs](#) (1), (2)
 - [ustawienie_typów_metadanych](#) (1), (2)
 - [ustawienie_minimalnego_wieku](#) (1), (2)
 - [ustawienie_czasu_ruchu_wstrzymania](#)
 - [ustawienie_serii_wielu_stron](#) (1), (2)
 - [ustawienie_folderu_probleków](#) (1), (2)
 - [ustawienie_metody_sortowania](#) (1), (2)
 - [ustawienie_trybu_drzewa](#) (1), (2)
 - [ustawienie_typów](#) (1), (2)
 - [ustawienie_czasu_oczekiwania](#)
 - [ustawienie_czasów_oczekiwania](#)

N

- Nazwa
 - [DCOSetupNode, właściwości](#)
- [nawigowanie](#)
- elementy nawigacyjne
 - [uzyskiwanie dostępu do hierarchii środowiska wykonawczego](#)

- NewDictionary, działanie (1), (2)
- NewDocument, działanie (1), (2)
- NewLine, działanie (1), (2)
- NormalizeCCO, działanie (1), (2)
- NumOfChildren
 - [DCO, metody](#)
- NumOfDictionaries
 - [DCOSetup, metody](#)
- NumOfNodes
 - [DCOSetup, metody](#)
- NumOfRules
 - [DCOSetupNode, metody](#)
- NumOfVariables
 - [DCOSetupNode, metody](#)
- NumOfVars
 - [DCO, metody](#)
- NumOfWords
 - [DCOSetup, metody](#)

O

- obiekt, interfejs
 - [DCO, metody](#)
 - [dokument, ścieżka hierarchii](#)
 - [FindRouteChild](#)
- obiekty
 - [powiązywanie, reguły](#)
- ObjectType
 - [DCO, metody](#)
 - [DCOSetupNode, właściwości](#)
- OCR/A
 - [pole wyboru, rozpoznanie](#)
- OCR_A, działania
 - [EnableEngineLogsOCR_A \(1\), \(2\)](#)
 - [OCRA_ConvertImage2BW \(1\), \(2\)](#)
 - [Rozpoznawanie](#)
 - [RecognizeBarcodeOCR_A \(1\), \(2\)](#)
 - [RecognizeFieldOCR_A \(1\), \(2\)](#)
 - [RecognizeFieldVoteOCR_A \(1\), \(2\)](#)
 - [RecognizePageFieldsOCR_A \(1\), \(2\)](#)
 - [RecognizePageOCR_A \(1\), \(2\)](#)
 - [RecognizeToALTOOCR_A](#)
 - [RecognizeToFileOCR_A](#)
 - [RecognizeToPDFOCR_A \(1\), \(2\)](#)
 - [ReleaseEngineOCR_A \(1\), \(2\)](#)
 - [RotateImageOCR_A \(1\), \(2\)](#)
 - [SetAutoRotationOCR_A \(1\), \(2\)](#)
 - [SetConfCalculation ParamsOCR_A \(1\), \(2\)](#)
 - [SetFastModeOCR_A \(1\), \(2\)](#)
- OCR_A, rozpoznanie
 - [automatyczne wykrywanie języka](#)
- OCR_J, działania
 - [InitializeEngine \(1\), \(2\)](#)
 - [Rozpoznawanie \(1\), \(2\)](#)

- ReleaseEngine (1), (2)
- OCR_N, działania
 - RecognizePageFieldsOCR_N (1), (2)
 - RecognizePageOCR_N (1), (2)
- OCR_S, działania
 - RecognizeDocToPDF (1), (2)
 - RecognizeFieldOCR_S (1), (2)
 - RecognizeFieldVoteOCR_S (1), (2)
 - RecognizeOM_OCR_S
 - RecognizePageFields 2CCO_OCR_S (1), (2)
 - RecognizePageFieldsOCR_S (1), (2)
 - RecognizePageOCR_S (1), (2)
 - RecognizePageOCR_S_2TextFile
 - RecognizePageOCR_S_2TextFile
 - RecognizePageOCR_S_Legacy
 - RecognizeToFile_OCR_S (1), (2)
 - RecognizeToPDF (1), (2)
 - RotateImage (1), (2)
 - SetEngineTimeout (1), (2)
 - SetFastTradeOffOCR_S (1), (2)
 - SetLegacyDecomposition OCR_S
 - SetLegacyDecompositionOCR_S
 - SetOutOfProcessLoggingOCR_S
- OCR_SR, działania
 - Rozpoznawanie
 - RecognizeFieldOCR_S (1), (2)
 - RecognizeFieldVoteOCR_S (1), (2)
 - RecognizePageFieldsOCR_S (1), (2)
 - RecognizePageOCR_S (1), (2)
 - RecognizeToFile_OCR_S (1), (2)
 - RecognizeToPDFOCR_S (1), (2)
 - RotateImageExOCR_S
 - RotateImageOCR_S (1), (2)
 - SetEngineTimeoutOCR_S (1), (2)
- OCRA_ConvertImage2BW, działanie (1), (2)
- OMR
 - metody
- OMR, pola
 - tworzenie reguły rozpoznawania
- OMR, strefy
 - rozpoznawanie, pola wyboru
- OpenConnection, działanie (1), (2), (3), (4)
- OpenDatabase, działanie (1), (2)
- Otwieranie zadanie wsadowego
 - Datacap Desktop
- OpenTextFaxServer, działania
 - Łączenie (1), (2)
 - ConnectOnConnectionError
 - ConnectOnFaxImportError
 - ContinueOn ConnectionError
 - Rozłączanie (1), (2)
 - ImportFaxes (1), (2)
 - SendAsFax (1), (2)

- SetAbortTimeout (1), (2)
- SetFaxRemovalAfterImport (1), (2)
- SetInputFolder (1), (2)
- SetMaxNumberOfFaxes
- SetMaxNumberOfFaxes
- SetNumberOfRetries (1), (2)
- SetPollingInterval (1), (2)
- SetProcessedFaxesFolder (1), (2)
- SetProtocol (1), (2)
- SetRetryTimeout (1), (2)
- SetServerName (1), (2)
- SetUserID (1), (2)
- SetUserPassword (1), (2)
- SetWindowsAuthentication (1), (2)
- działanie
 - Rulerunner
- rozpoznawanie znaczników optycznych
- Oracle, baza danych
 - planowanie
 - wymagania wstępne
 - wymagane oprogramowanie
- Bazy danych Oracle
 - konfigurowanie (1), (2)
 - konfigurowanie aplikacja dla
 - tworzenie
 - tworzenie pojedynczych użytkowników
 - planowanie
 - zapewnianie dostępu
 - uprawnienia zabezpieczeń
 - testowanie połączenia
- pozostałe informacje
 - uzyskiwanie dostępu
- Outlook, działania
 - OutlookMessageTo AttachmentOnly
 - OutlookMessageToImage AndAttachment
 - OutlookPrintQuality
 - OutlookTiffCompression
- OutlookPrintQuality, działanie
- OutlookTiffCompression, działanie
- Overlay, działanie (1), (2)
- nadpisywanie wartości poziomów ufności
- przegląd
 - Datacap
 - architektura
 - IBM Content Manager Connector, działania
- OverwriteExistingFiles
- OverwriteExistingFiles, działanie

P

- PadZone, działanie (1), (2)
- strona, dane
 - odczytywanie
- strona, pliki danych

- [spis treści](#)
 - [tworzenie](#)
- strona, pola
 - [określanie struktury](#)
- strona, odciski
 - [tworzenie](#)
- [PAGE_HEIGHT](#)
- [identyfikator strony](#)
 - [odcisk, ustalanie zgodności](#)
 - [ręczne \(1\), \(2\)](#)
 - [metody](#)
 - strony
 - [identyfikowanie, metody](#)
 - [dopasowywanie wzorca](#)
 - [ProtoId, klient WWW](#)
 - [oparte na strukturze](#)
 - [dopasowywanie tekstu \(1\), \(2\), \(3\)](#)
 - [korzystanie z klienta WWW AIndex](#)
- strona, zarządzanie
 - [w FastDoc \(1\), \(2\)](#)
- strona, status
 - [ustawienie](#)
- strona, typy
 - [tworzenie](#)
 - [tworzenie odcisków do](#)
 - [testowanie](#)
 - [identyfikowanie](#)
 - [identyfikowanie konkretnych pól](#)
 - [rejestrwanie obrazów](#)
 - [dopasowywanie wzorca](#)
 - [wersje](#)
- [PAGE_WIDTH](#)
- [PageID](#), czynność
 - [dodawanie rozgałęzienia warunkowego](#)
 - [TravelDocs](#)
- [PageIDByBCSep](#), działanie
 - [opis](#)
- [PageIDBySeqTypes](#), działanie
 - [opis](#)
- [PageIDByVariableChange](#), działanie
 - [opis](#)
- strony
 - [identyfikowanie](#)
 - [konfigurowanie](#)
 - [określanie struktury](#)
- [PageVariable_ExportValue](#), działanie (1), (2)
- panel, układ
 - [dostosowywanie](#)
- panele
 - [dodawanie kodu niestandardowego](#)
 - [przekształcanie \(1\), \(2\), \(3\), \(4\), \(5\), \(6\), \(7\)](#)
 - [tworzenie paneli Datacap Desktop \(1\), \(2\)](#)
 - [dostosowywanie do Datacap Navigator](#)

- generowanie układu, pliki XML (1), (2)
 - przeglądanie układu, pliki XML (1), (2)
- parametr, ustawienia
 - [Box Connector, działania](#)
 - [Documentum Connector, działania](#)
 - [Email Connector, działania](#)
 - [Fax Connector, działania](#)
 - [FileNet Image Services Connector, działania](#)
 - [FileNet P8 Connector, działania](#)
 - [IBM Content Manager Connector, działania](#)
 - [SharePoint Connector, działania](#)
- parametry
 - [ReportName](#)
 - [zapisywanie w pliku .app](#)
- Element nadrzędny
 - [DCO, metody](#)
- element nadrzędny, pole
 - [ustawianie wymaganych zmiennych](#)
- [Parse31aPhSig, działanie](#)
- [Parse58ainsnm, działanie](#)
- [Parse58binsnm, działanie](#)
- [Parse58cinsnm, działanie](#)
- [ParseConditionCodes, działanie](#)
- [ParseEPSDT, działanie](#)
- [ParseImageName, działanie \(1\), \(2\)](#)
- [ParseLastFirstIniNames, działanie](#)
- [ParseMultilineAddress, działanie \(1\), \(2\)](#)
- [ParseName, działanie \(1\), \(2\)](#)
- [ParseNDC, działanie](#)
- odciski częściowe
 - [usuwanie](#)
- hasła
 - [odwołanie z działań](#)
 - [zabezpieczanie \(1\), \(2\), \(3\), \(4\), \(5\), \(6\), \(7\), \(8\), \(9\)](#)
 - [zapisywanie](#)
 - [zapisywanie w pliku .app](#)
 - [rozwiązywanie problemów, niepowodzenie szyfrowania](#)
- [pat_RecogMatch_Id, działanie \(1\), \(2\)](#)
 - [opis](#)
 - [dopasowywanie wzorca](#)
- [pat_RegisterZones, działanie \(1\), \(2\)](#)
 - [dostosowywanie pozycji poszczególnych pól](#)
 - [dopasowywanie wzorca](#)
- [pat_ReleasePageAnchors, działanie \(1\), \(2\)](#)
- Ścieżka
 - [DCOSetup, właściwości](#)
- dopasowywanie wzorca
 - [dostosowywanie obrazów](#)
 - [dostosowywanie poszczególnych pozycji pól](#)
 - [dostosowywanie pozycji poszczególnych pól](#)
 - [wzorce zakotwiczenia](#)
 - rejestracja automatyczna
 - [korzystanie z działania FindFingerprint](#)

- [zagadnienia](#)
- [opis](#)
- [określanie pozycji pól w środowisku wykonawczym](#)
- [identyfikowanie stron](#)
- [rejestrwanie obrazów](#)
- [przegląd](#)
- [ustawianie poziomu ufności](#)
- [konfigurowanie obiektów zakotwiczonych](#)
- [oparte na tekście](#)
- [korzystanie z dopasowywania wzorca geometrycznego](#)
- [korzystanie z obiektów z wieloma zakotwiczeniami](#)
- [korzystanie z pat_RecogMatch_Id, działanie](#)
- [korzystanie z PatternMatch_Identify, działanie](#)
- [kiedy korzystać](#)
- [PatternConfidence](#)
- [PatternMatch](#)
- [PatternMatch, działania](#)
 - [MatchPattern \(1\), \(2\)](#)
 - [pat_RecogMatch_Id \(1\), \(2\)](#)
 - [pat_RegisterZones \(1\), \(2\)](#)
 - [pat_ReleasePageAnchors \(1\), \(2\)](#)
 - [PatternMatch_Fingerprint \(1\), \(2\)](#)
 - [PatternMatch_Identify \(1\), \(2\)](#)
 - [PatternMatch_PageType \(1\), \(2\)](#)
 - [SetMatchConfidence \(1\), \(2\)](#)
- [PatternMatch_Fingerprint, działanie \(1\), \(2\)](#)
- [PatternMatch_Identify, działanie \(1\), \(2\)](#)
 - [opis](#)
 - [dopasowywanie wzorca](#)
- [PatternMatch_PageType, działanie \(1\), \(2\)](#)
- [PD](#)
- [Pdf, działania](#)
 - [PDFBitDepth \(1\), \(2\)](#)
 - [PDFCompression \(1\), \(2\)](#)
 - [PDFConversionMethod \(1\), \(2\)](#)
 - [PDFDocumentToImage \(1\), \(2\)](#)
 - [PDFGrayscale \(1\), \(2\)](#)
 - [PDFHorizontalResolution \(1\), \(2\)](#)
 - [PDFQuality \(1\), \(2\)](#)
 - [PDFVerticalResolution \(1\), \(2\)](#)
- [PDFBitDepth, działanie \(1\), \(2\)](#)
- [PDFCompression, działanie](#)
- [PDFCompression, działanie](#)
- [PDFConversion Mode, działanie](#)
- [PDFConversionMethod, działanie \(1\), \(2\)](#)
- [PDFConversionMode, działanie](#)
- [PDFDocumentTo Image, działanie](#)
- [PDFDocumentToImage, działanie \(1\), \(2\), \(3\)](#)
- [PdfFRE, działania](#)
 - [PDFConversion Mode](#)
 - [PDFConversionMode](#)
 - [PDFDocumentTo Image](#)
 - [PDFDocumentToImage](#)

- [PDFImage Compression](#)
 - [PDFImageCompression](#)
 - [PDFImageFile Extension](#)
 - [PDFImageFile Resolution](#)
 - [PDFImageFileExtension](#)
 - [PDFImageFileResolution](#)
 - [PDFImageUse FastBinarization](#)
 - [PDFImageUseFastBinarization](#)
- [PDFGrayscale](#), działanie (1), (2)
- [PDFHorizontalResolution](#), działanie (1), (2)
- [PDFImage Compression](#), działanie
- [PDFImageCompression](#), działanie
- [PDFImageFile Extension](#), działanie
- [PDFImageFileExtension](#), działanie
- [PDFImageFileResolution](#), działanie
- [PDFImageFileResolution](#), działanie
- [PDFImageUse FastBinarization](#), działanie (1), (2)
- [PDFQuality](#), działanie (1), (2)
- [PDFVerticalResolution](#), działanie (1), (2)
- uprawnienia
 - [folder Datacap](#)
 - [nadawanie uprawnień do kont w Rulerunner](#)
 - [współużytkowanie, foldery](#)
- [PIC_ApplyPictureString](#), działanie (1), (2)
- [PIC_FilterFields](#), działanie (1), (2)
- [PIC_FormatFields](#), działanie (1), (2)
- [PIC_ReplaceBlankField](#), działanie
- [PIC_ReplaceBlankField](#), czynność
- [PIC_SetPictureCharacter](#), działanie (1), (2)
- [PIC_ValidateField](#), działanie (1), (2)
- Rysunek, działania
 - [PIC_ApplyPictureString](#) (1), (2)
 - [PIC_FilterFields](#) (1), (2)
 - [PIC_FormatFields](#) (1), (2)
 - [PIC_ReplaceBlankField](#) (1), (2)
 - [PIC_SetPictureCharacter](#) (1), (2)
 - [PIC_ValidateField](#) (1), (2)
- [PictureString](#)
 - [Konfiguracja DCO](#)
 - Weryfikacja, panel
 - [Datacap Web Client](#)
- [PilotMessage_Clear](#), działanie (1), (2)
- [PilotMessage_Set](#), działanie (1), (2)
- metoda oceny progu pikseli
 - [pole wyboru, rozpoznanie](#)
- rozpoznawanie progu pikseli
 - [sprawdzanie wartości](#)
 - [gęstość, wartości łańcuchów](#)
- Planowanie w systemie Datacap
 - [Uwierzytelnianie, system](#)
 - [Bazy danych](#)
 - [Domena, konta](#)
 - [Użytkownicy, grupy, stacje](#)

- planowanie
 - konta
 - uwierzytelnianie
 - bazy danych, użytkownicy
 - Datacap
 - użytkownicy i grupy
 - domeny
 - wymagania wstępne
 - skaner (1), (2), (3)
 - stacje
 - aktualizacja
 - użytkownicy i grupy
- Planowanie
- POLR, działania
 - CallPOLR (1), (2)
- PopulateFromField, działanie
- PopulateTaxType, działanie
 - opis
- PopulateWithResult, działanie (1), (2)
- PopulateZNField, działanie (1), (2)
- PopulateZNLineItemField, działanie (1), (2)
- PopulateZNLineItemFieldDynamic, działanie
- Uzupelnianie, obiekty konfiguracji
 - DCO, metody
 - SetupObject
- Portugalski
 - kody języków
- Pos<templateID>
- Pozycja
- POST, metoda
 - ChangeUserPassword
 - CopyFilesToCache
 - CreateBatch
 - DeleteBatches
 - Wykonywanie
 - Wykonywanie reguł w transakcji
 - Wykonywanie reguł na zadaniu wsadowym
 - Wylogowanie
 - Zalogowanie się
 - SaveBatchAttribute
 - SaveTask
 - SetFile
 - Obiekt strony nie został dodany do pliku strony
 - SetGroupPermissionList
 - SetUserPermissionList
 - Transaction.End
 - UploadFile
 - Obiekt strony dodany do pliku strony
- wymagania wstępne
 - konfiguracja
 - tworzenie raportów dla Report Viewer
 - DB2, baza danych
 - Documentum Connector, działania

- [Email Connector, działania](#)
- [Fax Connector, działania](#)
- [FileNet Image Services Connector, działania](#)
- [FileNet P8 Connector, działania](#)
- [Fingerprint, usługa](#)
- [IBM Content Manager Connector, działania](#)
- [instalacja](#)
- [instalowanie i konfigurowanie](#)
- [Microsoft .Net Framework 3.5.1](#)
- [Microsoft .Net Framework 4.0](#)
- [Microsoft aspnet_regiis.exe](#)
- [Microsoft Internet Information Services](#)
- [Microsoft SQL Server, baza danych](#)
- [Oracle, baza danych](#)
- [Report Viewer, raporty](#)
- [Rulerunner, konfiguracja](#)
- [SharePoint Connector, działania](#)
- [problem, zadania wsadowe](#)
 - [monitorowanie \(1\), \(2\)](#)
 - [raportowanie w \(1\), \(2\)](#)
- [problem, określanie](#)
 - [przeszukiwanie baz wiedzy](#)
- [ProcessChangeBatchStatus, działanie \(1\), \(2\)](#)
- [ProcessChangeBatchStatusOrder, działanie](#)
- [ProcessChangeBatchStatusTaskOrder, działanie](#)
- [ProcessChildren](#)
- [ProcessChildren, działanie \(1\), \(2\), \(3\)](#)
- [ProcessClearAuditTable, działanie \(1\), \(2\)](#)
- [ProcessClearDebugTable, działanie \(1\), \(2\)](#)
- [ProcessDeleteBatches, działanie \(1\), \(2\)](#)
- [ProcessDeleteBatchesEx, działanie](#)
- [ProcessDeleteBatchesEX, działanie](#)
- [ProcessInjectBatches, działanie \(1\), \(2\)](#)
- [ProcessMoveBatches, działanie \(1\), \(2\)](#)
- [ProcessMoveBatchesEx, działanie](#)
- [ProcessMoveBatchesEX, działanie](#)
- [ProcessMoveDBRecords, działanie \(1\), \(2\)](#)
- [ProcessResetPendingOrNotify, działanie](#)
- [ProcessRunSqlQuery, działanie \(1\), \(2\)](#)
- [ProcessRunSqlQueryEx, działanie](#)
- [Prof_Verify, czynność](#)
- [Ustawienia programu, plik](#)
 - [GET, metoda](#)
 - [GetProgramFile](#)
- [cele platformy projektu](#)
 - [ustawienia Report Viewer](#)
- [projekty](#)
 - [dodawanie elementów sterujących ReportViewer](#)
 - [budowanie](#)
- [PropagateToAltText, działanie \(1\), \(2\)](#)
- [ProtoId, klient WWW](#)
 - [konfigurowanie](#)
- [PUT, metoda](#)

- [GrabBatch](#)
- [GrabNextPendingBatch OnJobTaskList](#)
- [ReleaseBatch](#)
- [SetPageFileName](#)

Q

- [Zapytanie, działanie konfiguracyjna](#)
- Zapytanie, działania konfiguracyjne
 - [QueryClear \(1\), \(2\)](#)
 - [QuerySetAge \(1\), \(2\)](#)
 - [QuerySetBatchRange \(1\), \(2\)](#)
 - [QuerySetBranch \(1\), \(2\)](#)
 - [QuerySetDateFormat \(1\), \(2\)](#)
 - [QuerySetDateRange \(1\), \(2\)](#)
 - [QuerySetDateTimeFormat \(1\), \(2\)](#)
 - [QuerySetGeneric \(1\), \(2\)](#)
 - [QuerySetJobID \(1\), \(2\)](#)
 - [QuerySetOperator \(1\), \(2\)](#)
 - [QuerySetPriority \(1\), \(2\)](#)
 - [QuerySetSeparator \(1\), \(2\)](#)
 - [QuerySetStation \(1\), \(2\)](#)
 - [QuerySetStatus \(1\), \(2\)](#)
 - [QuerySetTaskID \(1\), \(2\)](#)
- [QueryClear, działanie \(1\), \(2\)](#)
- [QuerySetAge, działanie \(1\), \(2\)](#)
- [QuerySetBatchRange, działanie \(1\), \(2\)](#)
- [QuerySetBranch, działanie \(1\), \(2\)](#)
- [QuerySetDateFormat, działanie \(1\), \(2\)](#)
- [QuerySetDateRange, działanie \(1\), \(2\)](#)
- [QuerySetDateTimeFormat, działanie \(1\), \(2\)](#)
- [QuerySetGeneric, działanie \(1\), \(2\)](#)
- [QuerySetJobID, działanie \(1\), \(2\)](#)
- [QuerySetOperator, działanie \(1\), \(2\)](#)
- [QuerySetPriority, działanie \(1\), \(2\)](#)
- [QuerySetSeparator, działanie \(1\), \(2\)](#)
- [QuerySetStation, działanie \(1\), \(2\)](#)
- [QuerySetStatus, działanie \(1\), \(2\)](#)
- [QuerySetTaskID, działanie \(1\), \(2\)](#)

R

- szybkie tworzenie aplikacji
 - [FastDoc](#)
- rdlc, pliki
 - [modyfikowanie w serwerze SQL Server](#)
- Odczyt
 - [DCO, metody](#)
- [ReadBarCode, działanie \(1\), \(2\)](#)
- [ReadBarcodeBP, działanie](#)
- [ReadBarCodeBP, działanie](#)
- [ReadCurrentObjVariable, działanie \(1\), \(2\)](#)
- [ReadDCOSetup, działanie](#)
- [ReadFieldValue, działanie \(1\), \(2\)](#)

- ReadFPXMLZones, działanie (1), (2)
- odczyt, pliki
 - DCO, metody
 - Read
- odczyt, pliki konfiguracji DCO
 - DCO, metody
 - ReadSetup
- Readlock
 - DCOSetup, metody
- ReadOnly
 - Konfiguracja DCO
 - Weryfikacja, panel
 - Datacap Desktop
 - Datacap Web Client
- ReadPageSetup, działanie
- ReadPageVariableValue, działanie (1), (2)
- ReadSetup
 - DCO, metody
 - DCOSetup, metody
- ReadZones, działanie (1), (2)
- ReadZonesFPX, działanie (1), (2)
- Recog Shared, działania
 - SetupAutomaticRetry
- Recog_Shared, działania
 - AnalyzeImage (1), (2)
 - CCONormalization_OFF (1), (2)
 - CreateTextFile (1), (2)
 - IsBlankPage (1), (2)
 - RecogContinueOnFailure (1), (2)
 - RecogOMRThresh (1), (2)
 - RecogOMRThreshold (1), (2)
 - RegisterPageFields (1), (2)
 - Release_Image
 - ReleaseImage
 - RotateTio (1), (2)
 - SetAdjustFieldToChars (1), (2)
 - SetFingerprintRecogPriority (1), (2)
 - SetFullPageRecogArea (1), (2)
 - SetOutOfProcessRecogTimeout
 - SetOutOfProcessRecogTimeout
 - SetRecogFailureRetryDelay (1), (2)
 - SnapCCOtoDCO (1), (2)
 - SnapDCOtoCCO (1), (2)
 - SnapFieldtoChars (1), (2)
 - UseOutOfProcessRecog (1), (2)
- RecogContinueOnFailure, działanie (1), (2)
- rozpoznawanie
 - automatyczne wykrywanie języka
 - konfigurowanie we wzorcu formularza, aplikacje
- rozpoznanie, języki
 - ustawienia
- rozpoznawanie, reguła
 - tworzenie w łącznej siatce elementów

- rozpoznanie, strefy
 - [tworzenie dla odcisków](#)
 - [definiowanie](#)
 - [określanie dla TravelDocs](#)
 - [zapisywanie informacji](#)
 - [korzystanie z odcisków](#)
- Rozpoznawanie, działanie (1), (2), (3), (4)
- Rozpoznanie pól OMR, reguła
 - [dodawanie do hierarchii dokumentu](#)
 - [aktualizowanie przed korzystaniem z RecogOMRThreshold](#)
- Strona rozpoznawania, reguła
 - [aktualizowanie \(1\), \(2\)](#)
- Strona rozpoznawania, zestaw reguł
 - [aktualizowanie do ręcznej identyfikacji strony](#)
- [RecognizeBarcodeOCR_A](#), działanie (1), (2)
- [RecognizeDocToPDF](#), działanie (1), (2)
- [RecognizeFieldICR_C](#), działanie (1), (2)
- [RecognizeFieldICR_CEx](#), działanie
- [RecognizeFieldOCR_A](#), działanie (1), (2)
- [RecognizeFieldOCR_S](#), działanie (1), (2), (3), (4)
- [RecognizeFieldOCR_S_Legacy](#), działanie
- [RecognizeFieldsICR_P](#), działanie
- [RecognizeFieldVoteICR_C](#), działanie (1), (2)
- [RecognizeFieldVoteOCR_A](#), działanie (1), (2)
- [RecognizeFieldVoteOCR_S](#), działanie (1), (2), (3), (4)
- [RecognizeOM_OCR](#), działanie
- [RecognizePageFields2CCO_OCR_S](#), działanie
- [RecognizePageFields2CCO_ICR_C](#), działanie
- [RecognizePageFields2CCO_OCR_S](#), działanie
- [RecognizePageFieldsICR_C](#), działanie (1), (2)
- [RecognizePageFieldsICR_P](#), działanie
- [RecognizePageFieldsOCR_A](#), działanie (1), (2)
- [RecognizePageFieldsOCR_N](#), działanie (1), (2)
- [RecognizePageFieldsOCR_S](#), działanie (1), (2), (3), (4)
- [RecognizePageICR_C](#), działanie (1), (2)
- [RecognizePageOCR_A](#), działanie (1), (2)
- [RecognizePageOCR_N](#), działanie (1), (2)
- [RecognizePageOCR_S_2TextFile](#), działanie (1), (2)
- [RecognizePageOCR_S](#), działanie (1), (2), (3), (4)
- [RecognizePageToPDFICR_C](#), działanie (1), (2)
- [RecognizeToALTOOCR_A](#), działanie
- [RecognizeToFile_OCR_S](#), działanie (1), (2), (3), (4)
- [RecognizeToFileOCR_A](#), działanie
- [RecognizeToPDF](#), działanie (1), (2)
- [RecognizeToPDFOCR_A](#), działanie (1), (2)
- [RecognizeToPDFOCR_S](#), działanie (1), (2)
- [RecognizeOMRThresh](#), działanie (1), (2)
- [RecognizeOMRThreshold](#), działanie (1), (2)
- [RecogStatus](#)
- [RecogType](#)
- Redagowanie, działanie (1), (2)
- RedactByRegEx, działanie (1), (2)
 - [opis](#)

- RedactField, działanie
 - opis
- RedactParameter, działanie
- RedactParameters, działanie
- reference
 - zmienne właściwe dla aplikacji
 - Fingerprint Maintenance Tool
 - Medical Claims Capture
 - 5010, formularz dla instytucji
 - 5010, formularz firmowy
 - parametr inteligentny, specjalne zmienne
 - zmienne standardowe
- RegExFind, działanie (1), (2)
- RegExFind_InBlock, działanie
- RegExFind_InZone, działanie (1), (2)
- RegExFindNext, działanie (1), (2)
- RegExFindNext_InBlock, działanie
- RegExFindNext_InZone, działanie
- kody regionów
 - ustawienie
- Rejestrowanie strony
 - korzystanie z ręcznego zakotwiczenia
- RegisterPage, działanie (1), (2)
- RegisterPageFields, działanie (1), (2)
- klucze rejestru
 - punkty końcowe transakcji
- wyrażenia regularne
 - korzystanie za pomocą dopasowywania tekstu
- Wydawanie zadania wsadowego
 - PUT, metoda
 - ReleaseBatch
- ReleaseEngine, działanie (1), (2)
- ReleaseEngineOCR_A, działanie (1), (2)
- ReleaseImage, działanie (1), (2)
- zdalne skanowanie
 - Datacap Web Client
- zdalne skanowanie, klient
 - konfigurowanie (1), (2)
 - konfigurowanie Rulerunner
 - tworzenie czynności skanowania
 - modyfikowanie skrótu do opcji weryfikacji
 - czynności skanowania
 - skanowanie zadań wsadowych
 - Przesyłanie, czynność
 - przesyłanie zadań wsadowych
 - weryfikowanie zadania wsadowego
 - Web Job CreateDocs, czynność
- Zdalne skanowanie wirtualne
- RemoveDocumentStructure, działanie
 - opis
- usuwanie
 - Datacap (1), (2)
 - konta

- Application Pools
 - tworzenie kopii zapasowej, skróty w aplikacji
 - tworzenie kopii zapasowej, aplikacje niestandardowe
 - usuwanie skrótów do programu
 - usuwanie kluczy rejestru
 - usuwanie pozostałych folderów
 - usuwanie stron
- Usuwanie
- RenameFile, działanie (1), (2)
- ReplaceChars, działanie (1), (2)
- ReplaceMetadata
- ReplaceMetadata, działanie
- ReplaceValueAtPosition, działanie (1), (2)
- treść raportu
 - [zmienianie wielkości](#)
- raport, wykresy
 - [dodawanie](#)
- raport, nazwy kolumn
 - [zmiana](#)
- raport, kolumny
 - [dodawanie agregatów](#)
 - [tworzenie wyrażeń](#)
 - [zmienianie wielkości](#)
- raport, dane
 - [zmienianie nazw pól](#)
 - [filtrowanie \(1\), \(2\)](#)
 - [generowanie](#)
 - [konfigurowanie filtrów](#)
- raport, pliki definicji
 - [kopiowanie](#)
- raport, wymiary
 - [zmienianie wielkości](#)
- raport, filtry
 - [tworzenie](#)
- raport, wiersze stopki
 - [dodawanie](#)
- raport, elementy
 - [grupowanie](#)
- raport, strony
 - [konfigurowanie](#)
- [Report Viewer](#)
 - [uzyskiwanie dostępu](#)
 - [dodawany puli aplikacji \(1\), \(2\)](#)
 - [dodawanie raportów do pulpitu Report Viewer](#)
 - [dodawanie adresu strony jako zaufanego \(1\), \(2\)](#)
 - [konfigurowanie \(1\), \(2\), \(3\), \(4\)](#)
 - [tworzenie konta domeny dla](#)
 - [tworzenie filtrów raportu](#)
 - [tworzenie raportów \(1\), \(2\)](#)
 - [dodawanie elementów sterujących ReportViewer](#)
 - [tworzenie projektów](#)
 - [kopiowanie plików definicji raportu](#)
 - [kopiowanie plików reports.xml](#)

- projektowanie raportów
 - modyfikowanie plików .rdlc na serwerze SQL Server
 - ustawianie celów platformy projektu
 - uruchamiania aplikacji Formularze w systemie Windows
 - przeglądanie raportów niestandardowych
- raporty niestandardowe (1), (2)
- dostosowywanie
- dostosowywanie raportów
 - dodawanie agregatów
 - dodawanie agregatów do stopek kolumn
 - dodawanie wykresów
 - dodawanie wierszy stopki
 - tworzenie wyrażeń
 - zmienianie nazw pól
 - zmienianie nazw kolumn w raportach
 - obliczanie wartości danych
 - filtrowanie danych raportów (1), (2)
 - formatowanie tabel
 - grupowanie elementów
 - modyfikowanie raportów Datacap
 - zmiana wielkości treści w raporcie
 - zmiana wielkości kolumn w raporcie
 - zmiana wielkości raportów
 - konfigurowanie filtrów
 - konfigurowanie stron
- włączanie uwierzytelniania ADSI
- włączanie uwierzytelniania LDAP
- instalowanie (1), (2), (3), (4), (5)
- logowanie
- modyfikowanie pulpitu Report Viewer
- przegląd (1), (2), (3)
- wymagania wstępne
- nazwy raportów (1), (2)
- konfigurowanie w środowisku serwera klienta
- konfigurowanie uprawnień dostępu
- konfigurowanie uprawnień współużytkowania
- konfigurowanie serwisu WWW (1), (2)
- korzystanie ze statystyk Datacap
- przeglądanie raportów (1), (2)
- Report Viewer, adresy
 - dodawanie strony jako zaufanej (1), (2)
- Report Viewer, projekty
 - budowanie
- Raportowanie, działania
 - ReportQueryTMUsage (1), (2)
 - ReportSetReportingTable (1), (2)
 - ReportSetUsageDBTable (1), (2)
- ReportQueryTMUsage, działanie (1), (2)
- raporty
 - dodawanie do pulpitu Report Viewer
 - zadanie wsadowe, status (1), (2)
 - kolumny
 - zmienianie wielkości

- tworzenie dla Report Viewer (1), (2)
 - dodawanie elementów sterujących ReportViewer
 - tworzenie projektów w Report Viewer
 - kopiowanie plików definicji raportu
 - kopiowanie plików reports.xml
 - projektowanie
 - modyfikowanie plików .rdlc na serwerze SQL Server
 - ReportViewer Design Wizard
 - ustawianie celów platformy projektu dla przeglądarki raportu
 - uruchamiania aplikacji Formularze w systemie Windows
 - przeglądanie raportów niestandardowych w Report Viewer
- dostosowywanie (1), (2), (3)
- dostosowywanie do Report Viewer
 - dodawanie agregatów
 - dodawanie agregatów do stopek kolumn
 - dodawanie wykresów
 - dodawanie wierszy stopki
 - tworzenie wyrażeń
 - zmienianie nazw kolumn
 - zmienianie nazw pól
 - obliczanie wartości danych
 - filtrowanie danych raportów (1), (2)
 - formatowanie tabel
 - grupowanie elementów
 - modyfikowanie typów dokumentów
 - zmienianie wielkości
 - zmienianie wielkości treści
 - zmiana wielkości kolumn
 - konfigurowanie filtrów
 - konfigurowanie stron
- wyświetlanie wybranych filtrów
- lokalizacja
- modyfikowanie pulpitu Report Viewer
- problem, zadania wsadowe (1), (2)
- Report Viewer (1), (2)
- stacja, czynność (1), (2)
- tłumaczenie
- przeglądanie (1), (2)
- przeglądanie w Report Viewer
- pliki reports.xml
 - kopiowanie
- ReportSetReportingTable, działanie (1), (2)
- ReportSetUsageDBTable, działanie (1), (2)
- ReportViewer, elementy sterujące
 - dodawanie do projektów w Report Viewer
- ReportViewer Design Wizard
 - projektowanie raportów w Report Viewer
- ReqConf
- wymagane oprogramowanie
 - DB2, baza danych
 - Microsoft SQL Server, baza danych
 - Oracle, baza danych
- ResetField, działanie (1), (2)

- ResetFieldVariables, działanie (1), (2)
- API REST, metody
 - [Datacap Web Services](#)
- przekształcenie zadania wsadowego
 - [korzystanie z AIndex](#)
 - [korzystanie z VeriFine](#)
- zwracanie, obiekt podrzędny, indeks
 - [DCO, metody](#)
 - [FindChildIndex](#)
- zwracanie, zmienne, indeks
 - [DCO, metody](#)
 - [FindVariable](#)
- Zwraca plik
 - GET, metoda
 - [GetFile \(1\), \(2\)](#)
- przegląd
 - układ, pliki XML (1), (2)
- RightTruncate, działanie (1), (2)
- RotateImage, działanie (1), (2)
- RotateImageOCR_A, działanie (1), (2)
- RotateImageOCR_S, działanie (1), (2)
- RotateTio, działanie (1), (2)
- trasa, dokumenty
 - [rozgałęzianie](#)
 - [dzielenie](#)
 - [używanie rozgałęzień i dzielenia](#)
- Ustalanie trasy, zestaw reguł
 - [konfigurowanie](#)
 - [aktualizowanie w celu podziału rozgałęzienia](#)
- rr_AbortBatch, działanie (1), (2)
- rr_Get, działanie (1), (2)
- rr_WriteNode, działanie (1), (2)
- rrAppend, działanie (1), (2)
- rrCompare, działanie (1), (2)
- [rrCompareCase, działanie](#)
- [rrCompareCaseLength, działanie](#)
- rrCompareNot, działanie (1), (2)
- [rrCompareNotCase, działanie](#)
- [rrCompareNotCaseLength, działanie](#)
- rrCopy, działanie (1), (2)
- rrPrepend, działanie (1), (2)
- RRS
 - [pliki dziennika](#)
- RRS, folder
 - [Ustawianie zabezpieczeń](#)
 - [konfigurowanie zabezpieczeń](#)
- RRS, plik dziennika
 - [przeglądanie](#)
- rrSet, działanie (1), (2)
- rrunner, działania
 - [AbortOnError \(1\), \(2\)](#)
 - [CheckAllIntegrity \(1\), \(2\)](#)
 - [CheckDocCount \(1\), \(2\)](#)

- CheckPageCount (1), (2)
- DebugMode_OFF (1), (2)
- DebugMode_ON (1), (2)
- GoToNextFunction (1), (2)
- [MessageID](#)
- [MessageIDParameter](#)
- PilotMessage_Clear (1), (2)
- PilotMessage_Set (1), (2)
- ProcessChildren (1), (2)
- rr_AbortBatch (1), (2)
- [rr_Ger](#)
- [rr_Get](#)
- rr_WriteNode (1), (2)
- rrAppend (1), (2)
- rrCompare (1), (2)
- [rrCompareCase](#)
- [rrCompareCaseLength](#)
- rrCompareNot (1), (2)
- [rrCompareNotCase](#)
- [rrCompareNotCaseLength](#)
- rrCopy (1), (2)
- rrPrepend (1), (2)
- rrSet (1), (2)
- SetBatchPriority (1), (2)
- SetOperatorID (1), (2)
- SetReturnValue (1), (2)
- SetStationID (1), (2)
- SetTaskStatus (1), (2)
- SkipChildren (1), (2)
- Status_Preserve_OFF (1), (2)
- Status_Preserve_ON (1), (2)
- NumberOfSplits_zadanie (1), (2)
- RaiseCondition_zadanie (1), (2)
- kanały informacyjne RSS
 - [rozwiązywanie problemów](#)
- Rtf, działania
 - RtfPrintQuality (1), (2)
 - RtfTiffCompression (1), (2)
 - RtfToImage (1), (2)
- RtfPrintQuality, działanie (1), (2)
- [RtfTiffCompression, działanie](#)
- [RtfTiffCompression, działanie](#)
- RtfToImage, działanie (1), (2)
- wykonywanie reguły
 - [na poziomie zadania wsadowego, przykład](#)
 - [strona, kolejność identyfikowania](#)
 - [na poziomie strony, przykład](#)
 - [podsumowanie](#)
 - [kolejność sprawdzania poprawności](#)
- RuleChildName
 - [DCOSetupNode, właściwości](#)
- Rulemanager, karta
 - [panele](#)

- RuleMaxNum
 - [DCOSetupNode, właściwości](#)
- RuleMinNum
 - [DCOSetupNode, właściwości](#)
- RuleObjectType
 - [DCOSetupNode, właściwości](#)
- RulePosition
 - [DCOSetupNode, właściwości](#)
- Rulerunner
 - [dodawanie grup](#)
 - [dodawanie stacji](#)
 - [konfiguracja, wymagania wstępne](#)
 - [konfigurowanie](#)
 - [Konfigurowanie uwierzytelniania](#)
 - [Lista kontrolna](#)
 - [konfigurowanie \(1\), \(2\), \(3\), \(4\), \(5\), \(6\), \(7\), \(8\)](#)
 - [konfigurowanie uwierzytelniania \(1\), \(2\), \(3\), \(4\), \(5\), \(6\)](#)
 - [Konfigurowanie w TravelDocs](#)
 - [konfigurowanie profili czynności](#)
 - [konfigurowanie przed uruchomieniem aplikacji](#)
 - [konfigurowanie przed uruchomieniem czynności wykonywanych w tle](#)
 - [konfigurowanie zabezpieczeń w folderze aplikacji](#)
 - [konfigurowanie zabezpieczeń w folderze Datacap](#)
 - [konfigurowanie zabezpieczeń w folderze RRS](#)
 - [konfigurowanie współużytkowania w folderze Datacap](#)
 - [konfigurowanie przed uruchomieniem profili czynności](#)
 - [konfigurowanie przed uruchomieniem czynności](#)
 - [tworzenie kont dla](#)
 - [Datacap, uwierzytelnianie](#)
 - [Datacap Server, uprawnienia](#)
 - [opis](#)
 - [zbieranie informacji konfiguracyjnych](#)
 - [nadawanie nazwy domeny](#)
 - [nadawanie nazwy grupy uprawnień](#)
 - [przyznawanie przywileju logowania jako usługi](#)
 - [Instalacja i konfiguracja](#)
 - [Lista kontrolna](#)
 - [przegląd instalacji](#)
 - [instalowanie \(1\), \(2\), \(3\), \(4\), \(5\)](#)
 - [instalowanie na serwerze Rulerunner](#)
 - [instalowanie komponentów oprogramowania innych firm](#)
 - [pliki dziennika](#)
 - [rejestrwanie](#)
 - [logowanie do Datacap Web Client](#)
 - [monitorowanie, zadania wsadowe \(1\), \(2\)](#)
 - [wykonywanie działań](#)
 - [przegląd \(1\), \(2\)](#)
 - [konfigurowanie zabezpieczeń](#)
 - [zamykanie oprogramowania Datacap](#)
 - [na jednym komputerze](#)
 - [konfiguracja wątku](#)
 - [rozwiązywanie problemów](#)
 - [włączanie rejestrowania przetwarzania w Rulerunner](#)

- [priorytet do sprawdzenia](#)
 - [uruchamianie usługi Rulerunner \(1\), \(2\), \(3\)](#)
 - [zatrzymywanie usługi Rulerunner](#)
 - [przeglądanie dzienników Windows Event Viewer](#)
 - [przeglądanie dzienników przetwarzania Rulerunner](#)
- Rulerunner, konto
 - Konfigurowanie uprawnień
 - [Lista kontrolna](#)
- Rulerunner, konta
 - [nadawanie uprawnień do Rulerunner](#)
- Rulerunner, aplikacja, czynności
 - Konfigurowanie
 - [Lista kontrolna](#)
- Rulerunner, plik dziennika
 - [analizowanie](#)
 - [wyłączanie](#)
- Rulerunner, rejestrowanie
 - [włączanie](#)
- Rulerunner, dzienniki
 - [Ustawianie rejestrowania wg aplikacji](#)
 - [Ustawianie rejestrowania wg czynności](#)
- Rulerunner Manager
 - [Rejestrowanie, karta](#)
 - [Rulerunner, Logowanie, karta](#)
 - [Rulerunner, karta](#)
 - [Ustawienia, karta](#)
 - [Przeływ pracy:Zadanie:Czynność, karta](#)
- Rulerunner, usługa
 - [Uwierzytelnianie](#)
 - [Włączenie rejestrowania dla rozwiązywania problemów](#)
 - [Rulerunner](#)
 - [Rulerunner Manager](#)
 - [pojedynczy wątek](#)
- Rulerunner, usługa, właściwości
 - [Przywilej logowania jako usługi](#)
- reguły (1), (2)
 - [dodawanie istniejących do nowych stron](#)
 - [przypisywanie domyślnych do nowych pól](#)
 - [przypisywanie domyślnych do nowych stron](#)
 - [powiązanie z hierarchią dokumentu](#)
 - [dotaczanie do hierarchii dokumentu](#)
 - [sprawdzanie wiarygodności biznesowej](#)
 - [tworzenie w celu usunięcia elementów nieprzypisanych do wierszy](#)
 - [kolejność wykonywania](#)
- zestaw reguł, konfiguracja
 - FastDoc
 - [Datacap Server, tryb](#)
- zestawy reguł (1), (2)
 - [modyfikowanie](#)
- Wykonywanie zadania wsadowego
 - PUT, metoda
 - [GrabBatch](#)
- Uruchamianie klienta, aplikacja

- [Lista kontrolna](#)
 - [Datacap Web Client](#)
- uruchamianie, zestawy reguł
 - automatycznie
 - [Windows Task Scheduler](#)
 - ręczne
 - [Datacap Maintenance Manager](#)
- Uruchamianie czynności
 - [Datacap Desktop](#)
- [RunDecisionPlanCC](#), działanie (1), (2)
- [RunDecisionPlanForBlocksCC](#), działanie (1), (2)
- [RunDecisionPlanForTextCC](#), działanie (1), (2)
- [RunFlexIDPanel](#), działanie
 - [opis](#)
- uruchamianie zadania wsadowego w przepływie pracy (1), (2), (3)
- uruchamianie zadania wsadowego w przepływie pracy
 - [przygotowywanie](#)
- [Uruchamianie czynności poprawek za pomocą Datacap Desktop](#)
- Uruchamianie czynności skanowania
 - [Datacap Desktop](#)
 - [Datacap Web Client](#)
- [Uruchamianie czynności VScan za pomocą Datacap Desktop](#)
- Uruchamianie sprawdzania poprawności WWW
 - [TravelDocs](#)
- środowisko wykonawcze, folder zadań wsadowych
 - [identyfikator zadania wsadowego](#)
 - [spis treści](#)
 - [lokalizacja](#)
- środowisko wykonawcze, hierarchia zadań wsadowych
 - [zdefiniowane](#)
 - [powiązane z hierarchią dokumentu](#)
- Środowisko wykonawcze, hierarchia zadań wsadowych
 - [API](#)
- środowisko wykonawcze, plik danych
 - [aktualizowanie rozpoznany tekst](#)
- środowisko wykonawcze, pozycje pól
 - [określanie dla dopasowania wzorca](#)
- środowisko wykonawcze, hierarchia
 - uzyskiwanie dostępu (1), (2), (3), (4)
- [Środowisko wykonawcze, plik danych ze strony](#)
- środowisko wykonawcze, strony
 - [sprawdzanie poziomów ufności](#)
- Rosyjski
 - [kody języków](#)

S

- Aplikacja przykładowa
 - [TravelDocs](#)
- aplikacje przykładowe
 - [otwieranie](#)
- przykładowe odciski
 - [udoskonalanie](#)
- [przykładowe obrazy](#)

- Zapisywanie, atrybuty zadań wsadowych
 - POST, metoda
 - [SaveBatchAttribute](#)
- Zapisywanie, lista uprawnień grupy
 - POST, metoda
 - [SetGroupPermissionList](#)
- Zapisywanie, lista uprawnień użytkownika
 - POST, metoda
 - [SetUserPermissionList](#)
- [SaveAsCurrentObjVariable](#), działanie (1), (2)
- [SaveAsPageVariable](#), działanie (1), (2)
- [SaveDocToFolder](#), działanie (1), (2)
- [SaveFieldsText](#), działanie
- [SaveFilePathAsVariable](#), działanie (1), (2)
- [SaveObjectVariable](#), działanie (1), (2)
- [SaveToFile](#), działanie (1), (2)
- skanowanie
 - [zdalnie](#)
- Skan
 - [Datacap Navigator](#)
 - [lokalny](#)
 - [zdalnie](#)
 - uruchamianie, panel (1), (2)
 - [wirtualny](#)
- skan, działanie (1), (2)
- Skan, działanie (1), (2)
- [skan, czynność](#)
 - [tworzenie skrótu](#)
- Skan, czynność
 - [Medical Claims](#)
- [ScanDetails](#), działanie (1), (2)
- [ScanDetailsByLines](#), działanie (1), (2)
- [ScanDetailsByVSpace](#), działanie (1), (2)
- [ScanLineItem](#), działanie (1), (2)
- [ScanLineItemDynamic](#), działanie (1), (2)
- skaner, konfiguracja
 - [FastDoc](#)
- [skanowanie](#)
 - [dokumentu w formie elektronicznej](#)
 - [skaner lokalny](#)
 - [wirtualny](#)
- Skanowanie stron z wydrukowanych dokumentów
 - [Datacap Desktop](#)
- [Skanowanie, zdalnie](#)
- skanowanie, Datacap Web Client
 - [zdalne](#)
- [ScanRT](#), działanie (1), (2)
- skany
 - [weryfikowanie, czynność](#)
 - [weryfikowanie](#)
- [ScanSrcChannel](#)
- [ScanSrcFileName](#)
- [ScanSrcInputFolder](#)

- [ScanSrcPath](#)
- [ScanSrcSubFolder](#)
- scenariusz
 - [Obrazowanie](#)
 - [Zlecenie, automatyzacja przetwarzania](#)
- [SearchInSubdirectory](#), działanie (1), (2)
- zabezpieczenia
 - [klucze szyfrowania](#) (1), (2), (3), (4), (5), (6), (7), (8), (9)
 - [konfigurowanie dla Rulerunner](#)
 - [rozwiązywanie problemów](#)
 - [rozwiązywanie problemów, niepowodzenie szyfrowania](#)
- [Security Access Manager](#)
 - [Datacap Web Client](#)
- grupa uprawnień, nazwa
 - [pobieranie do Rulerunner, uwierzytelnianie](#)
- zabezpieczenia, klucze
 - [generowanie w FastDoc](#)
- zabezpieczenia, uprawnienia
 - bazy danych
 - [DB2](#)
 - [Oracle](#)
 - [SQL Server](#)
 - [konfigurowanie w Report Viewer](#)
- [SELECT](#)
 - [Konfiguracja DCO](#)
 - Weryfikacja, panel
 - [Datacap Desktop](#)
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
- [SelectParentBlock](#), działanie
- [SelectParentBlockOuterType](#), działanie
- [SelectParentBlockType](#), działanie
- [SelectSnippet](#), działanie (1), (2)
- [SendAsFax](#), działanie (1), (2)
- [SendEMail](#), działanie (1), (2)
- [SendOutlookNotification](#), działanie (1), (2)
- usługi
 - [Datacap Web Client, przesyłanie, usługa](#)
- [ustawienie_czasu_oczekiwania](#), działanie (1), (2)
- [ustawienie, uprawnienie do współużytkowania kont](#)
 - [Datacap Maintenance Manager](#)
- [set_AltConfidenceString](#)
 - [DCO, metody](#)
- [set_AltTex](#)
 - [DCO, metody](#)
- [set_CharConfidence](#)
 - [DCO, metody](#)
- [set_CharValue](#)
 - [DCO, metody](#)
- [ustawienie_kopiowania_folder](#), działanie (1), (2)
- [ustawienie_usuwania_pustych_folderów](#), działanie (1), (2)
- [set_DictionaryName](#)
 - [DCOSetup, metody](#)

- Ustawienie separatora w dokumencie
 - [ISIS, skaner](#)
- ustawienie_folder, działanie (1), (2)
- [ustawianie ogólnych punktów zatrzymania](#)
- ustawienie_sprawdzania_poprawności_obrazu, działanie (1), (2)
- ustaw lokalizację
 - Datacap Server (1), (2)
 - plik datacap.xml (1), (2)
- Ustawianie rejestrowania wg aplikacji i czynności
 - [Rulerunner, dzienniki](#)
- set_max_docs, działanie (1), (2)
- ustawienie_typów_metadanych, działanie (1), (2)
- ustawienie_minimalnego_wieku, działanie (1), (2)
- [ustawienie_czasu_ruchu_wstrzymania](#), działanie
- ustawienie_serii_wielu_stron, działanie (1), (2)
- Ustawienie, status następnego zadania wsadowego oczekującego na utworzenie
 - PUT, metoda
 - [GrabNextPendingBatch OnJobTaskList](#)
- set_OMRValue
 - [DCO, metody](#)
- ustawienie_folderu_probleatów, działanie (1), (2)
- set_RuleChildName
 - [DCOSetupNode, metody](#)
- set_RuleMaxNumber
 - [DCOSetupNode, metody](#)
- set_RuleMinNumber
 - [DCOSetupNode, metody](#)
- set_RuleObjectType
 - [DCOSetupNode, metody](#)
- set_RulePosition
 - [DCOSetupNode, metody](#)
- ustawienie_metody_sortowania, działanie (1), (2)
- ustawienie_trybu_drzewa, działanie (1), (2)
- ustawienie_typów, działanie (1), (2)
- set_Value
 - [DCOSetup, metody](#)
- set_Variable
 - [DCO, metody](#)
 - [DCOSetupNode, metody](#)
- set_VariableName
 - [DCOSetupNode, metody](#)
- set_VariableValue
 - [DCOSetupNode, metody](#)
- ustawienie_czasu_oczekiwania, działanie (1), (2)
- set_Word
 - [DCOSetup, metody](#)
- SetAbortTimeout, działanie (1), (2)
- SetAdjustedWidth, działanie (1), (2)
- SetAdjustFieldToChars, działanie (1), (2)
- SetAdminDB, działanie (1), (2)
- SetAlternateImageNames, działanie (1), (2)
- SetApplication, działanie (1), (2)
- SetApplicationID, działanie (1), (2)

- [SetAttachment, działanie](#)
- [SetAttachment, działanie](#)
- [SetAutoRotationOCR_A, działanie \(1\), \(2\)](#)
- [SetBackgroundImage, działanie \(1\), \(2\)](#)
- [SetBatchPriority, działanie \(1\), \(2\)](#)
- [SetBlindCarbonCopyRcpts, działanie \(1\), \(2\)](#)
- [SetCarbonCopyRcpts, działanie \(1\), \(2\)](#)
- [SetChrominanceFactor, działanie \(1\), \(2\)](#)
- [SetConf, działanie](#)
- [SetConfCalculationParamsOCR_A, działanie](#)
- [SetConfCalculationParamsOCR_A, działanie](#)
- [SetCSV, działanie \(1\), \(2\)](#)
- [SetDCOStatus, działanie \(1\), \(2\)](#)
- [SetDCOType, działanie](#)
- [SetDecisionPlan, działanie](#)
- [SetDecisionPlanCC, działanie](#)
- [SetDecisionPlanFieldsCC, działanie \(1\), \(2\)](#)
- [SetDeleteOriginal, działanie \(1\), \(2\)](#)
- [SetDetailsAndLineitemPairFPX, działanie](#)
- [SetDirectoryFPX, działanie \(1\), \(2\)](#)
- [SetDitheringBackground, działanie \(1\), \(2\)](#)
- [SetDocStatus, działanie \(1\), \(2\)](#)
- [SetDocumentType, działanie \(1\), \(2\)](#)
- [SetDynamicDetailZones, działanie \(1\), \(2\)](#)
- [SetElementSeparator, działanie \(1\), \(2\)](#)
- [SetEmailBody, działanie \(1\), \(2\)](#)
- [SetEngineDB, działanie \(1\), \(2\)](#)
- [SetEngineTimeout, działanie \(1\), \(2\)](#)
- [SetEngineTimeoutOCR_S, działanie \(1\), \(2\)](#)
- [SetEOL, działanie \(1\), \(2\)](#)
- [SetEOL_CRLF, działanie \(1\), \(2\)](#)
- [SetExportPath, działanie \(1\), \(2\)](#)
- [SetExtensionName, działanie \(1\), \(2\)](#)
- [SetFastMode, działanie \(1\), \(2\)](#)
- [SetFastModeOCR_A, działanie \(1\), \(2\)](#)
- [SetFastTradeOffOCR_S, działanie \(1\), \(2\)](#)
- [SetFaxRemovalAfterImport, działanie \(1\), \(2\)](#)
- [SetFileName, działanie \(1\), \(2\)](#)
- [SetFileReadOnly, działanie \(1\), \(2\)](#)
- [SetFill, działanie \(1\), \(2\)](#)
- [SetFilter_HostName, działanie \(1\), \(2\)](#)
- [SetFilter_PageType, działanie \(1\), \(2\)](#)
- [SetFingerprint, działanie \(1\), \(2\), \(3\)](#)
- [SetFingerprintDir, działanie](#)
- [SetFingerprintFailureThreshold, działanie \(1\), \(2\)](#)
- [SetFingerprintFolder, działanie \(1\), \(2\)](#)
- [SetFingerprintRecogPriority \(1\), \(2\)](#)
- [SetFingerprintSearchArea, działanie \(1\), \(2\)](#)
- [0SetFingerprintWebServiceURL, działanie \(1\), \(2\)](#)
- [SetFixedLength, działanie \(1\), \(2\)](#)
- [SetFldConfidence, działanie \(1\), \(2\)](#)
- [SetFontName, działanie \(1\), \(2\)](#)
- [SetFontSize, działanie \(1\), \(2\)](#)

- [SetFormType, działanie](#)
- [SetFullPageRecogArea, działanie \(1\), \(2\)](#)
- [SetGrayScale, działanie \(1\), \(2\)](#)
- [SetHaloBackground, działanie \(1\), \(2\)](#)
- [SetIgnoreFieldStatus, działanie \(1\), \(2\)](#)
- [SetImageType, działanie \(1\), \(2\)](#)
- [SetInputFolder, działanie \(1\), \(2\)](#)
- [SetIsOverrideable, działanie \(1\), \(2\)](#)
- [SetJustified, działanie \(1\), \(2\)](#)
- [SetKnowledgeBaseCC, działanie \(1\), \(2\)](#)
- [SetLabels, działanie](#)
 - [opis](#)
- [SetLanguageCC, działanie \(1\), \(2\)](#)
- [SetLegacyDecomposition OCR_S, działanie](#)
- [SetLegacyDecompositionOCR_S, działanie](#)
- [SetListenerURLCC, działanie \(1\), \(2\)](#)
- [SetLuminanceFactor, działanie \(1\), \(2\)](#)
- [SetMailServer, działanie \(1\), \(2\)](#)
- [SetMailSourceFolder, działanie \(1\), \(2\)](#)
- [SetMatchConfidence, działanie \(1\), \(2\)](#)
- [SetMaxCharacterHeightAVG, działanie \(1\), \(2\)](#)
- [SetMaxCharacterHeightTMM, działanie \(1\), \(2\)](#)
- [SetMaxImageFiles, działanie \(1\), \(2\)](#)
- [SetMaxNumberOfFaxes, działanie](#)
- [SetMaxNumberOfFaxes, działanie](#)
- [SetMaxOffset, działanie \(1\), \(2\)](#)
- [SetMaxTolerantDistance, działanie](#)
- [SetMinimumConfidence, działanie](#)
- [SetMinimumConfidenceBP, działanie](#)
- [SetMultiPageTiff, działanie \(1\), \(2\)](#)
- [SetNamePattern](#)
- [SetNamePattern, działanie](#)
- [SetNumberOfRetries, działanie \(1\), \(2\)](#)
- [SetOMR_Separator, działanie \(1\), \(2\)](#)
- [SetOpaque, działanie \(1\), \(2\)](#)
- [SetOperatorID, działanie \(1\), \(2\)](#)
- [SetOriginalTIF, działanie](#)
- [SetOutOfProcess RecogTimeout, działanie](#)
- [SetOutOfProcessRecogTimeout, działanie](#)
- [SetPageFingerprintID, działanie \(1\), \(2\)](#)
- [SetPageStatus, działanie \(1\), \(2\)](#)
- [SetPageTemplateID, działanie \(1\), \(2\)](#)
- [SetPageType, działanie \(1\), \(2\)](#)
- [SetPassword, działanie \(1\), \(2\)](#)
- [SetPicChar, działanie \(1\), \(2\)](#)
- [SetPollingInterval, działanie \(1\), \(2\)](#)
- [SetPosition](#)
 - [DCO, metody](#)
- [SetPostalDBPathICR_P, działanie \(1\), \(2\)](#)
- [SetProblemValue, działanie \(1\), \(2\)](#)
- [SetProblemValueCC, działanie \(1\), \(2\)](#)
- [SetProcessedFaxesFolder, działanie \(1\), \(2\)](#)
- [SetProfileString, działanie \(1\), \(2\)](#)

- SetProtocol, działanie (1), (2)
- SetRecipients, działanie (1), (2)
- SetRecogFailureRetryDelay, działanie (1), (2)
- SetRect, działanie (1), (2)
- SetRetryTimeout, działanie (1), (2)
- SetReturnValue, działanie (1), (2)
- SetSearchArea, działanie (1), (2)
- [SetSender](#), działanie
- SetServer, działanie (1), (2)
- SetServerName, działanie (1), (2)
- [SetSignatureReferenceFolderPath](#), działanie
- SetSortOrder, działanie (1), (2)
- SetSourceDirectory, działanie (1), (2)
- SetSpaceFill, działanie (1), (2)
- SetStation, działanie (1), (2)
- SetStationID, działanie (1), (2)
- SetStickyNo, działanie (1), (2)
- SetSubject, działanie (1), (2)
- SetTableName, działanie (1), (2)
- SetTaskStatus, działanie (1), (2)
- SetTemplateDir, działanie (1), (2)
- SetTIFFCompression, działanie (1), (2)
- ustawianie, dane pól znaków alternatywnych
 - [AltText](#)
 - [DCO](#), właściwości
- [Ustawianie IdentityDatacap Web Client dla puli aplikacji](#)
- [ustawianie znaków zatrzymania](#)
- ustawianie, dane znaków
 - [DCO](#), metody
 - [set_AltTex](#)
- ustawianie, wartość ufności dla danych dla znaków
 - [ConfidenceString](#)
 - [DCO](#), właściwości
- ustawianie, wartość dla danych dla znaków
 - [CharValue](#)
 - [DCO](#), właściwości
- ustawianie, pozycja obiektu dla znaków
 - [DCO](#), metody
 - [SetPosition](#)
- ustawianie, poziom ufności dla wartości znaków
 - [CharConfidence](#)
 - [DCO](#), właściwości
- ustawianie, wartości dla znaków
 - [DCO](#), metody
 - [set_CharValue](#)
- ustawianie poziomu ufności dla znaków
 - [AltConfidenceString](#)
 - [DCO](#), właściwości
- ustawianie, poziomy ufności
 - [DCO](#), metody
 - [set_CharConfidence](#)
- ustawianie, nazwa pliku obrazu
 - [DCO](#), właściwości

- [ImageName](#)
- ustawianie, nazwa obiektu
 - [DCO, właściwości](#)
 - [Typ](#)
- ustawianie, unikalny identyfikator obiektu
 - [DCO, właściwości](#)
 - [ID](#)
- ustawianie, podstawowe dane znaków
 - [DCO, właściwości](#)
 - [Tekst](#)
- ustawianie, środowisko wykonawcze, pozycja pola
 - [DCO, metody](#)
 - [SetPosition](#)
- Ustawianie, zabezpieczenia
 - [RRS, folder](#)
- ustawianie, securityDatacap Web Services
 - [aplikacja foldery](#)
 - [folder współużytkowany](#)
- ustawianie, właściwość Status
 - [DCO, właściwości](#)
 - [Status](#)
- Konfigurowanie uprawnień
 - [Użytkownicy](#)
- Konfigurowanie zabezpieczeń
 - Datacap\aplikacja, folder
 - [Właściwości Datacap](#)
 - folder Datacap
 - [Właściwości Datacap](#)
- Konfigurowanie uprawnień do współużytkowania
 - folder Datacap
 - [Właściwości Datacap](#)
- ustawianie, zmienna
 - [DCO, właściwości](#)
 - [Zmienna](#)
- [ustawianie, zmienne w polach opcji i ubezpieczeń](#)
- ustawianie, nazwa pliku XML
 - [DCO, właściwości](#)
 - [XML](#)
- ustawienia
 - [Datacap Navigator](#)
 - [języki rozpoznania](#)
- [Settings.ini](#)
- [SetToDocIDMPTIFF, działanie \(1\), \(2\)](#)
- [Konfiguracja DCO](#)
 - dodawanie węzłów
 - [AddNode](#)
 - [DCOSetup, metody](#)
 - dodawanie reguł
 - [AddRule](#)
 - [DCOSetupNode, metody](#)
 - dodawanie zmiennych
 - [AddVariable](#)
 - [DCOSetupNode, metody](#)

- usuwanie reguł
 - [DCOSetupNode, metody](#)
 - [DeleteRule](#)
- usuwanie zmiennych
 - [DCOSetupNode, metody \(1\), \(2\)](#)
 - [DeleteVariable](#)
 - [DeleteVariableByName](#)
- pobieranie, nazwa słownika
 - [DCOSetup, metody](#)
 - [DCOSetup, właściwości](#)
 - [DictionaryName](#)
 - [get_DictionaryName](#)
- pobieranie, wartość kluczowa terminu w słowniku
 - [DCOSetup, właściwości](#)
 - [Value](#)
- pobieranie, wartości kluczowych terminów w słowniku
 - [DCOSetup, metody](#)
 - [get_Value](#)
- pobieranie, wartości słów w słowniku
 - [DCOSetup, metody](#)
 - [get_Word](#)
- pobieranie, atrybut max obiektów podrzędnych
 - [DCOSetupNode, metody](#)
 - [get_RuleMaxNumber](#)
- pobieranie, atrybut min obiektów podrzędnych
 - [DCOSetupNode, metody](#)
 - [get_RuleMinNumber](#)
- Pobieranie węzłów
 - [DCOSetup, metody](#)
 - [GetNode](#)
- Pobieranie węzłów wg nazw
 - [DCOSetup, metody](#)
 - [GetNodeByName](#)
- pobieranie, liczba słowników
 - [DCOSetup, metody](#)
 - [NumOfDictionaries](#)
- pobieranie, liczba węzłów
 - [DCOSetup, metody](#)
 - [NumOfNodes](#)
- pobieranie, liczba węzłów
 - [DCOSetupNode, metody](#)
 - [NumOfRules](#)
- pobieranie, liczba zmiennych
 - [DCOSetupNode, metody](#)
 - [NumOfVariables](#)
- pobieranie, liczba słów
 - [DCOSetup, metody](#)
 - [NumOfWords](#)
- pobieranie, typ obiektu
 - [DCOSetupNode, metody](#)
 - [get_RuleObjectType](#)
- pobieranie, ścieżka
 - [DCOSetup, właściwości](#)

- [Path](#)
- pobieranie, nazwa reguły
 - [DCOSetupNode, metody](#)
 - [get_RuleChildName](#)
- pobieranie, atrybut pos reguły
 - [DCOSetupNode, metody](#)
 - [dokument, integralność](#)
 - [get_RuleObjectType](#)
 - [weryfikowanie kolejności stron w dokumentach](#)
- pobieranie, obiektów SetupNode
 - [DCOSetupNode, metody](#)
 - [GetRule](#)
 - [identyfikowanie obiektów wg reguł](#)
- pobieranie, nazwy zmiennych
 - [DCOSetupNode, właściwości](#)
 - [identyfikowanie obiektów wg wartości w indeksie](#)
 - [VariableName](#)
- pobieranie, wartości zmiennych
 - [DCOSetupNode, metody \(1\), \(2\), \(3\)](#)
 - [DCOSetupNode, właściwości](#)
 - [get_Variable](#)
 - [get_VariableName](#)
 - [get_VariableValue](#)
 - [identyfikowanie obiektów wg wartości zmiennych \(1\), \(2\), \(3\), \(4\)](#)
 - [Zmienna](#)
- pobieranie, wartość słowa
 - [DCOSetup, właściwości](#)
 - [Word](#)
- wyszukiwanie obiektu podrzędnego
 - [DCOSetupNode, metody](#)
 - [FindRule](#)
- ustawianie, wartości kluczowe w słowniku
 - [DCOSetup, metody](#)
 - [set_Value](#)
- ustawianie, nazwa słownika
 - [DCOSetup, metody](#)
 - [set_DictionaryName](#)
- ustawianie, wartość kluczowa terminu w słowniku
 - [DCOSetup, właściwości](#)
 - [Value](#)
- ustawianie, wartości słów w słowniku
 - [DCOSetup, metody](#)
 - [set_Word](#)
- ustawianie, atrybut max obiektów podrzędnych
 - [DCOSetupNode, metody](#)
 - [set_RuleMaxNumber](#)
- ustawianie, atrybut min obiektów podrzędnych
 - [DCOSetupNode, metody](#)
 - [set_RuleMinNumber](#)
- ustawianie, typ obiektu
 - [DCOSetupNode, metody](#)
 - [set_RuleObjectType](#)
- ustawianie, ścieżka

- [DCOSetup, właściwości](#)
 - [Path](#)
 - ustawianie, nazwa reguły
 - [DCOSetupNode, metody](#)
 - [set_RuleChildName](#)
 - ustawianie, atrybut pos reguły
 - [DCOSetupNode, metody](#)
 - [określanie kolejności stron w dokumentach](#)
 - [dokument, integralność](#)
 - [set_RuleObjectType](#)
 - ustawianie, nazwy zmiennych
 - [DCOSetupNode, metody](#)
 - [identyfikowanie zmiennych wg wartości w indeksie](#)
 - [set_VariableName](#)
 - ustawianie, wartości zmiennych
 - [DCOSetupNode, metody \(1\), \(2\)](#)
 - [DCOSetupNode, właściwości](#)
 - [identyfikowanie obiektów wg wartości zmiennych \(1\), \(2\), \(3\)](#)
 - [set_Variable](#)
 - [set_VariableValue](#)
 - [VariableValue](#)
 - ustawianie, wartość słowa
 - [DCOSetup, właściwości](#)
 - [Word](#)
 - wyświetlanie okna dialogowego konfiguracji
 - [DCOSetup, metody](#)
 - [ShowSetupDialog](#)
- Konfigurowanie pliku DCO
 - Usuwanie węzłów
 - [DCOSetup, metody](#)
 - [DeleteNode](#)
 - Usuwanie węzłów wg nazwy
 - [DCOSetup, metody](#)
 - [DeleteNodeByName](#)
 - blokowanie
 - [DCOSetup, metody](#)
 - [ReadLock](#)
 - odczytywanie
 - [DCOSetup, metody](#)
 - [ReadSetup](#)
 - odblokowywanie
 - [DCOSetup, metody](#)
 - [Unlockit](#)
- [SetupDisconnectAll, działanie \(1\), \(2\)](#)
- [SetupNode](#)
 - [DCO, metody](#)
- [SetupObject](#)
 - [DCO, metody](#)
- [SetupOpenApplication, działanie \(1\), \(2\)](#)
- [SetupOpenApplicationEx, działanie \(1\), \(2\)](#)
- [SetupUpNode DCO](#)
 - [określanie kolejności obiektów podrzędnych](#)
 - [DCOSetupNode, właściwości](#)

- [RulePosition](#)
 - określanie kolejności reguł
 - [DCOSetupNode, właściwości](#)
 - [RulePosition](#)
 - pobieranie, nazwa obiektu
 - [DCOSetupNode, właściwości](#)
 - [Nazwa](#)
 - pobieranie, typ obiektu
 - [DCOSetupNode, właściwości \(1\), \(2\)](#)
 - [ObjectType](#)
 - [RuleObjectType](#)
 - pobieranie, atrybut max reguły
 - [DCOSetupNode, właściwości](#)
 - [RuleMaxNum](#)
 - pobieranie, atrybut min reguły
 - [DCOSetupNode, właściwości](#)
 - [RuleMinNum](#)
 - pobieranie, nazwa reguły
 - [DCOSetupNode, właściwości](#)
 - [RuleChildName](#)
 - Ustawianie nazwy obiektu
 - [DCOSetupNode, właściwości](#)
 - [Nazwa](#)
 - Ustawianie typu obiektu
 - [DCOSetupNode, właściwości \(1\), \(2\)](#)
 - [ObjectType](#)
 - [RuleObjectType](#)
 - Ustawianie atrybutu max reguły
 - [DCOSetupNode, właściwości](#)
 - [RuleMaxNum](#)
 - Ustawianie atrybutu min reguły
 - [DCOSetupNode, właściwości](#)
 - [RuleMinNum](#)
 - Ustawianie nazwy reguły
 - [DCOSetupNode, właściwości](#)
 - [RuleChildName](#)
- [SetUser](#), działanie (1), (2)
- [SetUserID](#), działanie (1), (2)
- [SetUserPassword](#), działanie (1), (2)
- [SetWindowsAuthentication](#), działanie (1), (2)
- [SetZeroFill](#), działanie (1), (2)
- Współużytkowanie, działania
 - [ExceptionSetFileTypes](#)
 - [ExceptionSetHandler](#)
 - [ExceptionSetTaskCondition](#)
 - [ExceptionSetVariableName](#)
 - [SetNamePattern](#)
- [zabezpieczenia folderu współużytkowanego](#)
 - [Datacap Web Services](#)
- [SharedRecognitionTools](#), działanie
 - [SharedRecognitionTools](#)
- [SharePoint](#)
 - [Uwierzytelnianie](#)

- SharePoint Connector
 - [przesyłanie plików, przykłady](#)
- SharePoint Connector, działania
 - [konfigurowanie](#)
 - [przegląd](#)
 - [ustawienia parametrów](#)
 - [wymagania wstępne](#)
- współużytkowanie, uprawnienia
 - [konfigurowanie Report Viewer](#)
- Współużytkowanie uprawnień
 - [Datacap Web Services](#)
- Skróty, administracja
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
- ShowChar
 - [Konfiguracja DCO](#)
 - Weryfikacja, panel
 - [Datacap Web Client](#)
- ShowSetupDialog
 - [DCOSetup, metody](#)
- sprawdzanie poprawności podpisu, działania
 - [CreateNew](#)
 - [SetMinimumConfidence](#)
 - [SetSignatureReferenceFolderPath](#)
 - [ValidateSignature](#)
- na jednym komputerze
 - [konfigurowanie](#)
 - [konfigurowanie Internet Explorer](#)
 - [instalowanie i konfigurowanie](#)
 - [instalowanie komponentów](#)
 - [Rulerunner \(1\), \(2\)](#)
 - [Rulerunner, usługa](#)
- Na jednym komputerze
 - [Instalowanie i konfigurowanie](#)
 - [Instalowanie i konfigurowanie, Datacap](#)
- na jednym komputerze, instalacja
 - [Datacap Web Client \(1\), \(2\)](#)
 - [FastDoc](#)
- na jednym komputerze, instalacja, konfiguracja Internet Explorer
 - [Datacap Web](#)
- Pojedyncze logowanie
 - [Datacap Navigator](#)
- SkipChildren, działanie (1), (2)
- [parametr inteligentny, specjalne zmienne](#)
- Parametr inteligentny, specjalne zmienne
 - [uzyskiwanie dostępu do informacji o zadaniu i czynności](#)
 - [uzyskiwanie dostępu do hierarchii środowiska wykonawczego](#)
 - [uzyskiwanie dostępu do pliku konfiguracyjnego aplikacji](#)
 - [różne](#)
- parametry inteligentne
 - [uzyskiwanie dostępu do hierarchii środowiska wykonawczego](#)
 - [ustawienia aplikacji](#)
 - [uzyskiwanie dostępu korzystając ze specjalnych zmiennych](#)

- [określanie poprawnej nazwy kluczowej](#)
 - [elementy](#)
 - [przegląd](#)
 - [struktura](#)
 - [korzystanie](#)
- SmartSQL, działanie (1), (2)
- SnapCCOtoDCO, działanie (1), (2)
- SnapDCOtoCCO, działanie (1), (2)
- SnapFieldtoChars, działanie (1), (2)
- oprogramowanie
 - [ponowne uruchamianie](#)
- oprogramowanie, raporty o kompatybilności produktu
 - [szczegółowe wymagania systemowe](#)
 - [informacje o końcu świadczenia usług](#)
 - [wymagania sprzętowe](#)
 - [obsługiwane tłumaczenia](#)
- [rozwiązania](#)
- [SourceFolder](#)
- [SourceFolder, działanie](#)
- SP_CreateFolder, działanie (1), (2)
- SP_Login, działanie (1), (2)
- SP_SetContentType, działanie (1), (2)
- SP_SetFileType, działanie (1), (2)
- SP_SetProperty, działanie (1), (2)
- SP_SetUploadMode, działanie (1), (2)
- SP_SetUrl, działanie (1), (2)
- SP_Upload, działanie (1), (2)
- [SP_Upload Dir](#)
- [SP_UploadDir, działanie](#)
- Hiszpański
 - [kody języków](#)
- warunki specjalne
 - [tworzenie zadań dla](#)
- zmienne specjalne
 - [uzyskiwanie informacji o zadaniu \(1\), \(2\)](#)
 - [uzyskiwanie innych informacji](#)
 - [uzyskiwanie informacji o czynności \(1\), \(2\)](#)
 - [uzyskiwanie dostępu do hierarchii środowiska wykonawczego \(1\), \(2\)](#)
- SPExport, działania
 - [SP_CreateFolder \(1\), \(2\)](#)
 - [SP_Login \(1\), \(2\)](#)
 - [SP_SetContentType \(1\), \(2\)](#)
 - [SP_SetFileType \(1\), \(2\)](#)
 - [SP_SetProperty \(1\), \(2\)](#)
 - [SP_SetUploadMode \(1\), \(2\)](#)
 - [SP_SetUrl \(1\), \(2\)](#)
 - [SP_Upload \(1\), \(2\)](#)
 - [SP_Upload Dir](#)
 - [SP_UploadDir](#)
- Podział, działania
 - [SplitBatch \(1\), \(2\)](#)
- [SplitBatch, działanie \(1\), \(2\)](#)
- [SplitFieldValueLeft, działanie \(1\), \(2\)](#)

- SplitFieldValuePreserveEnd, działanie (1), (2), (3)
- [SplitFieldValuePreserveStart](#), działanie
- SplitFieldValueRight, działanie (1), (2)
- SplitFileName, działanie (1), (2)
- SplitMultipageTiff, działanie (1), (2)
- SplitTIFFCompression, działanie (1), (2)
- dzielenie
 - [definiowanie warunków](#)
 - [opis](#)
 - [zgłaszanie flag warunków](#)
- SQL Server, bazy danych
 - [konfigurowanie](#)
 - [konfigurowanie aplikacja dla](#)
 - [tworzenie](#)
 - [uprawnienia zabezpieczeń](#)
 - [testowanie połączenia](#)
- SSL
 - [Datacap Web Services](#)
- SSO
- raporty standardowe
 - [przeglądanie](#)
- zmienne standardowe
 - [wszystkie typy obiektów](#)
 - [zadanie wsadowe, zmienne](#)
 - [dokument, zmienne](#)
 - [pole, zmienne](#)
 - [strona, zmienne](#)
- uruchamianie
 - [Datacap Server Service](#)
- Uruchomienie transakcji
 - GET, metoda
 - [Transakcja/Uruchom](#)
- Uruchamianie, panel do skanowania zdalnego (1), (2)
 - [uruchamianie reguł sprawdzania wiarygodności](#)
- uruchamianie
 - [Datacap Studio](#) (1), (2)
- stacja, czynność
 - [monitorowanie](#) (1), (2)
 - [raportowanie w](#) (1), (2)
- stacje
 - [planowanie](#)
- Statystyki, działania
 - [AddToDBTotals](#)
 - [CompareFieldsText](#)
 - [IsBatchAborted](#)
 - [SaveFieldsText](#)
- Status
 - [DCO, właściwości](#)
- STATUS
- kody statusu
 - [interpretowanie](#)
- Status_Preserve_OFF, działanie (1), (2)
- Status_Preserve_ON, działanie (1), (2)

- wartości statusu
 - [pole](#)
 - [strona](#)
- Sticky
 - [Konfiguracja DCO](#)
 - Weryfikacja, panel
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
- Zatrzymanie
 - [Datacap Web Client, przesyłanie, usługa](#)
- Zapisywanie, aplikacja, ustawienia
 - [Application Manager](#)
- łańcuchy
 - [lokalizowanie za pomocą dopasowywania tekstu](#)
- [StripTrailingAlpha, działanie](#)
- [dokumenty ustrukturyzowane](#)
- [SumFields, działanie \(1\), \(2\)](#)
- [obsługa](#)
- narzędzia pomocy technicznej
 - [IBM Support Assistant \(ISA\)](#)
- obsługiwane kody języków
 - [Chiński](#)
 - [Holenderski](#)
 - [Wschodnioeuropejski](#)
 - [Angielski](#)
 - [Francuski](#)
 - [Niemiecki](#)
 - [Włoski](#)
 - [Portugalski](#)
 - [Rosyjski](#)
 - [Hiszpański](#)
 - [Szwedzki](#)
- [SwapImages, działanie \(1\), \(2\)](#)
- Szwedzki
 - [kody języków](#)
- [SwitchMMDD, działanie \(1\), \(2\)](#)
- system, konto
 - [dodawanie do grupy administracyjnej](#)
- system, wydajność
 - [monitorowanie](#)

T

- tabele
 - [formatowanie](#)
- [TargetFolder](#)
- [TargetFolder, działanie](#)
- task
 - [konfigurowanie](#)
 - [tworzenie](#)
- czynność, informacje
 - [uzyskiwanie dostępu \(1\), \(2\)](#)
- Czynność, lista
 - [Datacap Navigator](#)

- Task_NumberOfSplits, działanie (1), (2)
- [czynność, profile](#)
 - [konfigurowanie w Rulerunner](#)
- Task_RaiseCondition, działanie (1), (2)
- czynności
 - [tworzenie](#)
 - [opis](#)
 - [pliki dziennika](#)
- [TEMPLATE IMAGE](#)
- [TemplateID](#)
- test
 - [baza danych, połączenia](#)
- [Test, karta](#)
- Testowanie instalacji
 - Lista kontrolna
 - [Fingerprint, usługa](#)
- Testowanie, konfiguracja
 - [Datacap Web Client](#)
- Tekst
 - [DCO, właściwości](#)
 - [Środowisko wykonawcze DCO](#)
 - [Konfiguracja DCO](#)
 - Weryfikacja, panel
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
- Tekst, działanie (1), (2)
- [dopasowanie wzorca opartego na tekście](#)
- dopasowywanie tekstu
 - [identyfikowanie stron \(1\), \(2\)](#)
 - [ograniczenia](#)
 - [ustawienia narodowe, pola](#)
 - [lokalizowanie, dane \(1\), \(2\)](#)
 - [lokalizowanie, dane pola](#)
 - [lokalizowanie, proste łańcuchy](#)
 - [aktualizowanie pliku danych w środowisku wykonawczym](#)
 - [aktualizowanie aplikacji TravelDocs](#)
 - [dotaczanie reguł do hierarchii dokumentu](#)
 - [identyfikowanie nierozpoznanych stron](#)
 - [rozpoznawanie danych](#)
 - [uruchamianie zadania wsadowego w przepływie pracy](#)
 - [korzystanie z listy słów kluczowych](#)
 - [korzystanie z wyrażeń regularnych](#)
- Dopasowywanie tekstu
 - [strona, przykład zidentyfikowania](#)
- [dopasowywanie tekstu w celu lokalizowania pól](#)
- Tiff, działania
 - [SplitMultipageTiff \(1\), \(2\)](#)
 - [SplitTIFFCompression \(1\), \(2\)](#)
- TifMerge, działania
 - [TifMerge_CheckStatus](#)
 - [TifMerge_ExportToBatchDir \(1\), \(2\)](#)
 - [TifMerge_MergeImages \(1\), \(2\)](#)
 - [TifMerge_MyImage \(1\), \(2\)](#)

- [TifMerge_Preserve Compression](#)
 - [TifMerge_PreserveCompression](#)
 - [TifMerge_SetFileName \(1\), \(2\)](#)
 - [TifMerge_SetFilePath \(1\), \(2\)](#)
- [TifMerge_CheckStatus, działanie](#)
- [TifMerge_ExportToBatchDir, działanie \(1\), \(2\)](#)
- [TifMerge_MergeImages, działanie \(1\), \(2\)](#)
- [TifMerge_MyImage, działanie \(1\), \(2\)](#)
- [TifMerge_PreserveCompression, działanie \(1\), \(2\)](#)
- [TifMerge_SetFileName, działanie \(1\), \(2\)](#)
- [TifMerge_SetFilePath, działanie \(1\), \(2\)](#)
- [TimeStampField, działanie \(1\), \(2\)](#)
- [TM524, działania](#)
- TMA, uwierzytelnianie
 - [Użytkownik i grupa, uwierzytelnianie](#)
- [tmweb.net, pula aplikacji](#)
 - [konfigurowanie \(1\), \(2\)](#)
 - [Datacap Web Client, serwis \(1\), \(2\)](#)
- [Transakcja/Wykonywanie](#)
 - [klucze rejestru](#)
- [Transakcja, przechwytywanie](#)
 - [dodawanie dokumentów do zadań wsadowych](#)
 - [konfigurowanie, Datacap](#)
 - [konfigurowanie, IBM Case Manager](#)
 - [konfigurowanie, IBM Content Navigator \(1\), \(2\)](#)
 - [konfigurowanie stacji roboczej użytkownika](#)
 - [IBM Case Manager](#)
 - [IBM Content Navigator](#)
 - [przykładowe przepływy pracy](#)
 - [skanowanie w IBM Case Manager](#)
 - [skanowanie w IBM Content Navigator](#)
- [punkty końcowe transakcji](#)
 - [wyłączanie zabezpieczeń](#)
 - [uruchamianie reguł](#)
- [TransformLI, działanie](#)
- [tłumaczenie](#)
 - [aplikacje](#)
 - [raporty](#)
- [obsługiwane tłumaczenia](#)
 - [oprogramowanie, raporty o kompatybilności produktu](#)
- [dokumenty podrzędne](#)
 - [strony opcjonalne](#)
 - [strony wymagane](#)
- [TravelDocs](#)
 - [dodawanie nowych stron z siatkami elementów w wierszach](#)
 - [automatyczne przetwarzanie w tle](#)
 - [analizowanie dziennika Rulerunner](#)
 - [definiowanie czynności wykonywanych w tle](#)
 - [wyłączanie dziennika Rulerunner](#)
 - [włączanie rejestrowania Rulerunner](#)
 - [uruchamianie zadania wsadowego w przepływie pracy](#)
 - [konfigurowanie czynności wykonywanych w tle](#)
 - [konfigurowanie monitora zadań](#)

- Batch Profiler, czynność
- Konfigurowanie Rulerunner
- konfigurowanie profili czynności
- tworzenie, strefy tekstu
- tworzenie hierarchii dokumentu
- tworzenie biblioteki odcisków początkowych
- opracowywanie wymogów biznesowych
- włączanie ręcznej identyfikacji strony
 - dodawanie funkcji
 - dodawanie rozgałęzienia warunkowego do czynności ID strony
 - konfigurowanie rozgałęzień
 - konfigurowanie zestawu reguł przekazywania
 - tworzenie zadania i czynności ManualPageID
 - wykrywanie danych na niezidentyfikowanej stronie
 - uruchamianie zadania wsadowego w przepływie pracy
 - aktualizowanie zestawu reguł strony rozpoznawania
- eksportowanie danych do bazy danych
- eksportowanie danych do pliku XML
- eksportowanie danych z siatki elementów w wierszach do pliku XML
 - dodawanie reguł do zestawu danych ExportXML
 - dołączanie reguł Export Other XML do hierarchii dokumentu
 - uruchamianie zadania wsadowego w przepływie pracy
- pole, kody statusu
- generowanie odcisków automatycznie
 - dodawanie zestawu reguł do profilu czynności Weryfikowania (1), (2)
 - przypisywanie reguły do każdego typu stron
 - tworzenie zestawu reguł AutoFingerprint
 - przeglądanie pliku dziennika RRS
 - uruchamianie zadania wsadowego w przepływie pracy
- interpretowanie, gęstość, wartości łańcuchów
- strona, kody statusu
- PageID, czynność
- przetwarzanie, zadania wsadowe
- rozpoznawanie, dane siatki elementów w wierszach
- przekazywanie w celu obsługi niepowodzeń związanych z dokumentami
 - konfigurowanie Rulerunner przed uruchomieniem CreateDocs
 - tworzenie czynności CreateDocs
 - przenoszenie zadania tworzenia dokumentu i sprawdzanie integralności
 - uruchamianie zadania wsadowego w przepływie pracy
- uruchamianie zadania wsadowego w przepływie pracy
- uruchamianie automatycznego przetwarzania w tle
- Uruchamianie czynności sprawdzania poprawności WWW
- Aplikacja przykładowa
- przykładowe obrazy
- skanowanie przykładowych obrazów
- oddzielanie dokumentu z głównego rozgałęzienia
 - przypisywanie podziału zadań wsadowych, reguła
 - przekazywanie rozdzielonego dokumentu do przetłózonego (1), (2), (3), (4)
 - uruchamianie zadania wsadowego w przepływie pracy
 - aktualizowanie przekazywanie, zestaw reguł
- przechodzenie przez zadanie wsadowe
- testowanie czynności eksportu (1), (2)
- aktualizowanie przed skorzystaniem z opcji dopasowywania tekstu

- dołączanie reguł do hierarchii dokumentu
 - identyfikowanie nierozpoznanych stron
 - rozpoznawanie danych
 - uruchamianie zadania wsadowego w przepływie pracy
 - Przesyłanie, czynność
 - korzystanie z dopasowywania wzorca geometrycznego
 - przeglądanie plików zadań wsadowych w środowisku wykonawczym
 - uruchamianie zadania wsadowego w przepływie pracy
 - konfigurowanie strefy zakotwiczenia
 - aktualizowanie PageID, reguła
 - sprawdzanie poprawności kosztu lotu
 - sprawdzanie poprawności danych siatki elementów w wierszach
 - Weryfikowanie, czynność (1), (2)
 - VScan, czynność
 - Web VScan
- TrimSpaces, działanie
 - opis
- rozwiązywanie problemów (1), (2)
 - uwierzytelnianie
 - kontakt z IBM Software Support
 - niepowodzenie szyfrowania
 - klucze szyfrowania
 - FastDoc
 - słabe rozpoznawanie kodów kreskowych
 - typ dokumentu nie jest przypisywany automatycznie
 - błąd podczas eksportu
 - dodatkowe dane w danych przechwytywania
 - dane indeksowe niewybrane w polu strefy
 - skaner nie znajduje się na liście
 - SharePoint, przesyłanie (1), (2), (3)
 - Fingerprint Maintenance Tool
 - katalog kopii zapasowej
 - plik FMT.log
 - pobieranie poprawek
 - Fix Central
 - import, niepowodzenie
 - pliki dziennika
 - Rulerunner
 - włączanie rejestrowania przetwarzania w Rulerunner
 - priorytet do sprawdzenia
 - uruchamianie usługi Rulerunner (1), (2), (3)
 - zatrzymywanie usługi Rulerunner
 - przeglądanie dzienników Windows Event Viewer
 - przeglądanie dzienników przetwarzania Rulerunner
 - przeszukiwanie baz wiedzy
 - zabezpieczenia
 - subskrybowanie w celu uzyskiwania informacji z działu wsparcia
- Rozwiązywanie problemów
 - Datacap Web Services
 - GrabBatch
 - ReleaseBatch
 - SetFile
 - SetPageFileName

- [UploadFile](#)
 - Włączenie, rejestrowanie
 - [Datacap Desktop](#)
 - [Datacap Server Service](#)
 - [Datacap Web Client](#)
 - [Datacap Web Services](#)
 - [FastDoc](#)
 - [Rulerunner, usługa](#)
- [TruncateFromEnd](#), działanie (1), (2)
- [TruncateFromStart](#), działanie
 - opis
- zaufane strony
 - dodawanie adresów Report Viewer jako (1), (2)
- TWAIN, konfigurowanie skanera
 - [FastDoc](#)
- weryfikacja metodą podwójnego wprowadzania danych
- Weryfikacja metodą podwójnego wprowadzania danych
- Txt, działania
 - [TxtFontName](#)
 - [TxtFontSize](#)
 - [TxtPrintQuality](#) (1), (2)
 - [TxtTiffCompression](#) (1), (2)
 - [TxtToImage](#) (1), (2)
- [TxtFontName](#), działanie
- [TxtFontSize](#), działanie
- [TxtPrintQuality](#), działanie (1), (2)
- [TxtTiffCompression](#), działanie
- [TxtTiffCompression](#), działanie
- [TxtToImage](#), działanie (1), (2)
- Typ
 - [DCO, właściwości](#)
- [TYPE](#)

U

- [Deinstalowanie](#)
- [Unlockit](#)
 - [DCOSetup, metody](#)
- [update](#)
 - aplikacje (1), (2)
 - bazy danych (1), (2)
- [UpdateCredentialList](#), działanie
- [UpdateDCOField](#), działanie (1), (2)
- [UpdateField](#), działanie (1), (2)
- [UpdateFieldWithBlock](#), działanie
- [UpdateFPStats](#), działanie (1), (2), (3)
- [UpdateKnowledgeBaseCC](#), działanie (1), (2)
- aktualizacje
 - subskrypcja
 - [rozwiązywanie problemów](#)
- [aktualizowanie odcisku automatycznego przed uruchomieniem FPXML](#)
- [aktualizowanie ManualPageID](#)
- [aktualizowanie, reguła Strony rozpoznania](#)
- [aktualizowanie aplikacji](#)

- aktualizacja
 - [Tworzenie kopii zapasowej wcześniejszych wersji](#)
- aktualizacje
 - [Datacap, oprogramowanie](#)
 - [przegląd](#)
- [Aktualizowanie](#)
- [Przesyłanie](#)
 - [Datacap Navigator](#)
- Przesyłanie pliku
 - POST, metoda
 - [Obiekt strony dodany do pliku strony](#)
 - [UploadFile](#)
- Przesyłanie, działanie (1), (2), (3)
- przysyłanie, pliki
 - POST, metoda
 - [SetFile](#)
- Przesyłanie, pliki
 - POST, metoda
 - [Obiekt strony nie został dodany do pliku strony](#)
 - [SetFile](#)
- [Upload_SetNumAttempts, działanie \(1\), \(2\)](#)
- Przesyłanie, czynność
 - [TravelDocs](#)
- [UploadSetDelay, działanie](#)
- [adresy URL do Datacap Navigator](#)
- [UseIndexes\)_ON, działanie](#)
- [UseIndexes_OFF, działanie \(1\), \(2\)](#)
- [UseIndexes_ON, działanie](#)
- [UseOutOfProcessRecog, działanie \(1\), \(2\)](#)
- dostęp użytkownika
 - [przypisywanie do baz danych](#)
 - [bazy danych](#)
- Użytkownik i grupa, uwierzytelnianie
 - [TMA, uwierzytelnianie](#)
- Użytkownik, uwierzytelnianie
 - [ADLDS, uwierzytelnianie](#)
 - [LLLDAP, uwierzytelnianie](#)
- Użytkownik, konfiguracja
 - [Datacap Web Client Configuration Tool](#)
- użytkownik, interfejs
 - [klonowanie, aplikacje](#)
 - [migrowanie, aplikacje](#)
 - [aktualizowanie, aplikacje](#)
- Użytkownik, lista uprawnień
 - GET, metoda
 - [GetUserPermissionList](#)
- użytkownik, ustawienia
 - [Datacap Navigator](#)
- Użytkownik, konfiguracja stacji roboczej
 - [Lista kontrolna](#)
- Użytkownicy
 - [Upewnianie się, że konta istnieją](#)
 - [Konfigurowanie uprawnień](#)

- Użytkownicy, grupy, stacje
 - [Planowanie w systemie Datacap](#)

V

- Sprawdzanie poprawności typu samochodu, reguła
 - [dodawanie do hierarchii dokumentu](#)
 - [tworzenie](#)
- Sprawdzanie poprawności pola waluty, reguła
 - [dodawanie do hierarchii dokumentu](#)
 - [tworzenie](#)
- Sprawdzanie poprawności kosztu lotu, reguła
 - [tworzenie](#)
- [ValidateNPI, działanie](#)
- [ValidateSignature, działanie](#)
- [ValidateVendor, działanie \(1\), \(2\)](#)
- sprawdzanie poprawności typu samochodu
 - [wyszukiwanie, baza danych](#)
- sprawdzanie poprawności ścieżek do obiektów
 - [DCO, metody](#)
 - [IsRoute](#)
- sprawdzanie poprawności
 - [pola waluty](#)
 - [łączna liczba sieci \(1\), \(2\), \(3\)](#)
 - [zarządzanie błędami](#)
 - [nadpisywanie niepowodzeń](#)
 - [konfigurowanie we wzorcu formularza, aplikacje](#)
 - [konfigurowanie we wzorcu uczenia, aplikacje](#)
 - [korzystanie z zewnętrznych źródeł danych](#)
- sprawdzanie poprawności, niepowodzenia
 - [wyświetlanie operatorowi](#)
- sprawdzanie poprawności, reguły
 - [uruchamianie pól w panelu](#)
- [Statusy sprawdzania poprawności](#)
- Sprawdzanie poprawności, działania
 - [AddLeadingZeros \(1\), \(2\)](#)
 - [AddPaddingToEnd \(1\), \(2\)](#)
 - [AddPaddingToLeft \(1\), \(2\)](#)
 - [AddPaddingToRight \(1\), \(2\)](#)
 - [AddPaddingToStart \(1\), \(2\)](#)
 - [AddTrailingZeros \(1\), \(2\)](#)
 - [AllowOnlyChars \(1\), \(2\)](#)
 - [AppendFromField \(1\), \(2\)](#)
 - [AppendToField \(1\), \(2\)](#)
 - [AssignFieldDefault \(1\), \(2\)](#)
 - [Obliczanie \(1\), \(2\)](#)
 - [CalculateDateDifference \(1\), \(2\)](#)
 - [CalculateFields \(1\), \(2\)](#)
 - [CheckSubFields \(1\), \(2\)](#)
 - [CompareFields \(1\), \(2\)](#)
 - [ConvertFieldTo Currency](#)
 - [ConvertFieldToCurrency](#)
 - [ConvertToLowerCase \(1\), \(2\)](#)
 - [ConvertToUpperCase \(1\), \(2\)](#)

- CopyField (1), (2)
- CopyFieldToField (1), (2)
- DateStampField (1), (2)
- DeleteAllAlpha (1), (2)
- DeleteAllMiscChars (1), (2)
- DeleteAllNumeric (1), (2)
- DeleteAllPunct (1), (2)
- DeleteAllSysChars (1), (2)
- DeleteChildType (1), (2)
- DeleteLCSpaces (1), (2)
- DeleteParentObj (1), (2)
- DeleteSelectedChars (1), (2)
- EmptyFieldValue (1), (2)
- FailRuleSet (1), (2)
- FieldContainsValue (1), (2)
- FilterFieldSelectedChars (1), (2)
- FormatNumberToLocale (1), (2)
- GetJobID (1), (2)
- HasChildOfType (1), (2)
- InsertChars (1), (2)
- InsertDecimalPoint (1), (2)
- IsFieldCurrency (1), (2)
- IsFieldDate (1), (2)
- IsFieldDateEqualOrAfter (1), (2)
- IsFieldDateEqualOrBefore (1), (2)
- IsFieldDateUpToToday (1), (2)
- IsFieldDateWithinRange (1), (2)
- IsFieldDateWithinXDays (1), (2)
- IsFieldDateWithReformat (1), (2)
- IsFieldEmpty (1), (2)
- IsFieldFilled (1), (2)
- IsFieldGreaterOrEqual (1), (2)
- IsFieldHidden (1), (2)
- IsFieldLengthMax (1), (2)
- IsFieldLengthMin (1), (2)
- IsFieldLessOrEqual (1), (2)
- IsFieldMatching (1), (2)
- IsFieldPercent NonNumeric
- IsFieldPercentAlpha (1), (2)
- IsFieldPercentNonNumeric
- IsFieldPercentNumeric (1), (2)
- IsMatchingJobID (1), (2)
- IsMaxOMRChecked (1), (2)
- IsMinOMRChecked (1), (2)
- IsPatternInField (1), (2)
- IsSupportedImageFile (1), (2)
- IsThisFieldEmpty (1), (2)
- IsThisFieldFilled (1), (2)
- IsVariableEmpty (1), (2)
- IsVariableFilled (1), (2)
- LeftTruncate (1), (2)
- MessageBox (1), (2)
- ParseMultilineAddress (1), (2)

- ParseName (1), (2)
- ReadCurrentObjVariable (1), (2)
- ReadFieldValue (1), (2)
- ReadPageVariableValue (1), (2)
- ReplaceChars (1), (2)
- ReplaceValueAtPosition (1), (2)
- ResetField (1), (2)
- RightTruncate (1), (2)
- SaveAsCurrentObjVariable (1), (2)
- SaveAsPageVariable (1), (2)
- SetIsOverrideable (1), (2)
- SplitFieldValueLeft (1), (2)
- SplitFieldValuePreserveEnd (1), (2), (3)
- [SplitFieldValuePreserveStart](#)
- SplitFieldValueRight (1), (2)
- SumFields (1), (2)
- TimeStampField (1), (2)
- TrimSpaces (1), (2)
- TruncateFromEnd (1), (2)
- TruncateFromStart (1), (2)
- [ValProcedureCode](#), działanie
- [ValRequiredCode](#), działanie
- Wartość
 - [DCOSetup](#), właściwości
- ValueInField, działanie (1), (2)
- ValueInField_Fuzzy, działanie (1), (2)
- ValueInField_RegEx, działanie (1), (2)
- Zmienna
 - [DCO](#), właściwości
 - [DCOSetupNode](#), właściwości
- Variable_ExportValue, działanie (1), (2)
- Variable_IsValue, działanie (1), (2)
- VariableName
 - [DCOSetupNode](#), właściwości
- zmienne
 - [uzyskiwanie dostępu, ustawienia aplikacji](#)
 - [hr_locale](#)
 - [parametry inteligentne](#)
- VariableValue
 - [DCOSetupNode](#), właściwości
- weryfikowanie
 - [strony z siatkami elementów w wierszach](#)
 - [skany](#)
 - [pomijanie czynności](#)
- Weryfikacja, panel
 - [Datacap Navigator](#) (1), (2)
 - Konfiguracja DCO
 - [DataType](#)
 - [DICT](#)
 - [etykieta](#)
 - [Lookup](#)
 - [LookupEx](#)
 - [MaxLength](#)

- [MultiLine](#)
 - [MultiPunch](#)
 - [PictureString](#)
 - [Pozycja](#)
 - [ReadOnly](#)
 - [ReqConf](#)
 - [SELECT](#)
 - [ShowChar](#)
 - [Sticky](#)
 - [Tekst](#)
- [VeriFine, klient WWW](#)
 - [konfigurowanie \(1\), \(2\)](#)
 - [strony niestandardowe](#)
 - [przekształcanie zadania wsadowego](#)
- weryfikacja
 - [baza danych, połączenia](#)
- Weryfikacja, skrót
 - [modyfikowanie](#)
- Weryfikacja, czynność
 - [Medical Claims](#)
 - [TravelDocs \(1\), \(2\)](#)
- weryfikowanie
 - [AVerify, klient WWW](#)
 - [podwójnie ślepa próba](#)
 - [Fingerprint, usługa \(1\), \(2\)](#)
 - [wielokrotne wprowadzanie danych](#)
 - [podwójne wprowadzanie danych](#)
 - [korzystanie z AIndex](#)
 - [korzystanie z klienta WWW ImgEnter](#)
 - [korzystanie z VeriFine](#)
- Weryfikowanie, zadania wsadowe
 - [Datacap Web Client](#)
- Weryfikowanie statusu instalacji komponentów IIS
 - [Datacap Web Client](#)
 - [Weryfikowanie statusu instalacji komponentów IIS \(1\), \(2\), \(3\)](#)
- Weryfikowanie, strony
 - [Datacap Desktop](#)
- wersje
 - [strona, typy](#)
- Przeglądarka wirtualna
 - [aktualizowanie](#)
- Głosowanie, działania
 - [VoteFld \(1\), \(2\)](#)
- [VoteFld, działanie \(1\), \(2\)](#)
- [Vscan](#)
 - [przykładowe obrazy](#)
- [VScan](#)
 - [modyfikowanie, zestaw reguł](#)
- [Vscan, działania](#)
 - [AddDocument \(1\), \(2\)](#)
 - [CopyFile \(1\), \(2\)](#)
 - [DeleteImageFile \(1\), \(2\)](#)
 - [MoveImageFileToDirectory \(1\), \(2\)](#)

- Skanowanie (1), (2)
- SearchInSubdirectory (1), (2)
- SetAlternateImageNames (1), (2)
- SetFastMode (1), (2)
- SetImageType (1), (2)
- SetMailSourceFolder (1), (2)
- SetMaxImageFiles (1), (2)
- SetMultiPageTiff (1), (2)
- SetSortOrder (1), (2)
- SetSourceDirectory (1), (2)
- VScan, czynność
 - [Medical Claims](#)
 - [TravelDocs](#)
- VScanDatacap Studio
 - [generowanie zadania wsadowego](#)

W

- Web Client Configuration, narzędzie
 - [Datacap Web Client](#)
 - Opakowanie
 - [Lista kontrolna](#)
- Web Job CreateDocs, czynność
 - [tworzenie](#)
- Web Services, działanie
 - [WsClearHeaders \(1\), \(2\)](#)
 - [WsClearParameters \(1\), \(2\)](#)
 - [WsClearResultItems \(1\), \(2\)](#)
 - [WsEncodeParameter \(1\), \(2\)](#)
 - [WsGetFile \(1\), \(2\)](#)
 - [WsGetValues \(1\), \(2\)](#)
 - [WsSetCredentials \(1\), \(2\)](#)
 - [WsSetHeader \(1\), \(2\)](#)
 - [WsSetNamespace \(1\), \(2\)](#)
 - [WsSetParameter \(1\), \(2\)](#)
 - [WsSetResultItem \(1\), \(2\)](#)
 - [WsSetTimeout \(1\), \(2\)](#)
 - [WsUploadData \(1\), \(2\)](#)
 - [WsUploadFile \(1\), \(2\)](#)
- serwisy WWW
 - [konfigurowanie w Report Viewer \(1\), \(2\)](#)
- Web VScan
 - [TravelDocs](#)
- Windows 2008 IIS 7.5 Server
 - [Datacap Web Server, konfiguracja](#)
- Windows, konto
 - [programista](#)
- Windows, konta
 - [tworzenie dla Report Viewer](#)
- Windows, uwierzytelnianie
 - [konfigurowanie Datacap](#)
- Windows, Formularze, aplikacje
 - [tworzenie projektu](#)
 - [uruchamianie](#)

- Windows Server 2012 R2
 - [Datacap Web Server, konfiguracja](#)
- Opcje usługi hostingu w środowisku Windows
 - [Datacap Web Services](#)
- Windows, usługi
 - [Datacap Web Client, przesyłanie, usługa](#)
- Windows Task Scheduler
 - [automatyczne uruchamianie zestawów reguł](#)
- Word
 - [DCOSetup, właściwości](#)
- Word, działania
 - [WordDocumentToImage \(1\), \(2\)](#)
 - [WordDocumentToPdf](#)
 - [WordMonochromeQuality](#)
 - [WordPrintQuality \(1\), \(2\)](#)
 - [WordTiffCompression \(1\), \(2\)](#)
- WordDocumentToImage, działanie [\(1\)](#), [\(2\)](#)
- WordFind, działanie [\(1\)](#), [\(2\)](#)
- WordFind_InZone, działanie [\(1\)](#), [\(2\)](#)
- WordFind_Offset, działanie [\(1\)](#), [\(2\)](#)
- WordFindNext, działanie [\(1\)](#), [\(2\)](#)
- WordFindNext_InZone, działanie [\(1\)](#), [\(2\)](#)
- [WordPrintQuality, działanie](#)
- [WordPrintQuality, działanie](#)
- WordTiffCompression, działanie [\(1\)](#), [\(2\)](#)
- Przepływy pracy, zadania i czynności, administracja
 - [Datacap Navigator](#)
- Przepływy pracy, administracja
 - [Datacap Navigator](#)
 - [Datacap Web Client](#)
- [przepływy pracy](#)
 - [automatyczne przetwarzanie](#)
 - [tworzenie zadań](#)
 - [tworzenie czynności](#)
 - [Datacap Web Client Operations, karta](#)
 - [opis](#)
 - [przetwarzanie w Rulerunner](#)
 - [przekazywanie automatyczne](#)
 - [skrót](#)
- Przepływy pracy, zadania i czynności, administracja
 - [Datacap Web Client](#)
- Zapis
 - [DCO, metody](#)
- WriteErrorMessage, działanie [\(1\)](#), [\(2\)](#)
- WriteSetup
 - [DCO, metody](#)
 - [DCOSetup, metody](#)
- WriteZoneFPX, działanie [\(1\)](#), [\(2\)](#)
- WriteZonesFPX, działanie [\(1\)](#), [\(2\)](#)
- Pisanie, obiekty konfiguracji
 - [DCO, metody](#)
 - [WriteSetup](#)
- WsClearHeaders, działanie [\(1\)](#), [\(2\)](#)

- WsClearParameters, działanie (1), (2)
- WsClearResultItems, działanie (1), (2)
- WsEncodeParameter, działanie (1), (2)
- WsGetFile, działanie (1), (2)
- [WsGetValues, działanie](#)
- [WsGetValues, działanie](#)
- WsSetCredentials, działanie (1), (2)
- WsSetHeader, działanie (1), (2)
- WsSetNamespace, działanie (1), (2)
- WsSetParameter, działanie (1), (2)
- WsSetResultItem, działanie (1), (2)
- WsSetTimeout, działanie (1), (2)
- WsUploadData, działanie (1), (2)
- WsUploadFile, działanie (1), (2)

X

- XML
 - [DCO, właściwości](#)
- xml_CommitNode, działanie (1), (2)
- [xml_NewNode, działanie](#)
- [xml_NewNode, działanie](#)
- xml_SaveFile, działanie (1), (2)
- xml_SetAttributeValue, działanie (1), (2)
- xml_SetExportPath, działanie (1), (2)
- xml_SetFileName, działanie (1), (2)
- xml_SetNodeValue, działanie (1), (2)

Z

- Zip, działania
 - [ZipOverwrite \(1\), \(2\)](#)
 - [ZipPassword \(1\), \(2\)](#)
 - [ZipUnpack](#)
 - [ZipUnPack](#)
- [ZipOverwrite, działanie \(1\), \(2\)](#)
- [ZipPassword, działanie](#)
- [ZipPassword, działanie](#)
- [ZipUnpack, działanie](#)
- [ZipUnPack, działanie](#)
- [Zone_Offset](#)
- [ZoneBOTTOM_ImageBottom, działanie](#)
- [ZoneBOTTOM_LowerBound, działanie](#)
- [ZoneBOTTOM_UpperBound, działanie](#)
- [ZoneImage_SaveAs, działanie \(1\), \(2\)](#)
- [ZoneLEFT_ImageLeft, działanie \(1\), \(2\)](#)
- [ZoneLEFT_LeftBound, działanie](#)
- [ZoneLEFT_RightBound, działanie](#)
- [ZoneRIGHT_ImageRight, działanie \(1\), \(2\)](#)
- [ZoneRIGHT_LeftBound, działanie](#)
- [ZoneRIGHT_RightBound, działanie](#)
- Strefy, działania
 - [AdjustZonesToImageOffset \(1\), \(2\)](#)
 - [AnchorPage \(1\), \(2\)](#)

- CalculateLocalOffset (1), (2)
- CreateBlockCCO (1), (2)
- FindBlocks_WhiteSpace (1), (2)
- FindDataBlocks (1), (2)
- FindLineItems (1), (2)
- FindRegExBlocks (1), (2)
- FindZoneLineItems (1), (2)
- GetZoneText (1), (2)
- InheritParentPosition (1), (2)
- LoadBlockCCO (1), (2)
- LoadZones (1), (2)
- MCCOPositionAdjust (1), (2)
- MergeZones (1), (2)
- PadZone (1), (2)
- PopulateZNField (1), (2)
- PopulateZNLineItemField (1), (2)
- ReadZones (1), (2)
- RegisterPage (1), (2)
- ScanDetails (1), (2)
- ScanDetailsByLines (1), (2)
- ScanDetailsByVSpace (1), (2)
- ScanLineItem (1), (2)
- SetEOL (1), (2)
- SetEOL_CRLF (1), (2)
- ZoneBOTTOM_ImageBottom
- ZoneBOTTOM_LowerBound
- ZoneBOTTOM_UpperBound
- ZoneBOTTOM_ImageBottom
- ZoneBOTTOM_LowerBound
- ZoneBOTTOM_UpperBound
- ZoneImage_SaveAs (1), (2)
- ZoneLEFT_ImageLeft (1), (2)
- ZoneLEFT_LeftBound (1), (2)
- ZoneLEFT_RightBound (1), (2)
- ZoneRIGHT_ImageRight (1), (2)
- ZoneRIGHT_LeftBound (1), (2)
- ZoneRIGHT_RightBound (1), (2)
- ZoneTOP_ImageTop (1), (2)
- ZoneTOP_LowerBound (1), (2)
- ZoneTOP_UpperBound (1), (2)
- [Zones, karta](#)
- ZoneTOP_ImageTop, działanie (1), (2)
- ZoneTOP_LowerBound, działanie (1), (2)
- ZoneTOP_UpperBound, działanie (1), (2)
- [ZZoneLEFT_LeftBound, działanie](#)
- [ZZoneLEFT_RightBound, działanie](#)
- [ZZoneRIGHT_LeftBound, działanie](#)
- [ZZoneRIGHT_RightBound, działanie](#)