

Migrating from IBM InfoPrint Designer applications

Moving to advanced document communications with DocPath Boulder Suite

Bill Shaffer
Frazer Dixon

October 08, 2015
(First published October 08, 2015)

IBM® InfoPrint Designer debuted with IBM i Version 5 Release 1, provides the basic document design capabilities. The requirements for customer communications today are vastly more complex. DocPath, a long time IBM i partner, has developed the DocPath Boulder Suite software that enables you to easily migrate existing InfoPrint Designer applications to the advanced DocPath document design and production system. Moreover, once migrated, the applications can take advantage of the advanced functionality of the DocPath system. In addition, new document applications can also be built using advanced Microsoft® Windows® based design and tight production integration on IBM i. This article takes you through the migration process. A follow-on developerWorks article would explain the implementation of advanced design and production functions.

IBM released InfoPrint Designer in V5R1 to provide a tool for enhancing spooled file output on IBM i. There are a number of reasons to considering migrating InfoPrint Designer applications. First, InfoPrint Designer was designed to run on Windows, but is only supported up to Windows XP. Second, InfoPrint Designer was based on a subset of the Advanced Function Presentation (AFP) architecture that is very limited in addressing the more complex, variable designs needed today. Third, the ways in which businesses communicate with customers has changed dramatically, including the range of *channels* expected for those communications (that is, print, web, email, text, and social media). And finally, customers with InfoPrint Designer typically have many applications. The effort to convert them all would typically be sizable.

DocPath Boulder Suite provides an easy and seamless path for this conversion, enabling the use of an advanced and thoroughly modern customer communications platform going forward with:

- The latest architecture and technologies – (that is, Java™, HTML5, CCS)
- Single, consistent, intuitive user interface with fully graphical job workflow
- Wide language support, including the ability to use multiple languages within the same document
- Wide support for input data: XML, Tag Mode, Text, FCFC, variable and fixed record mode

- Wide support for print formats: PCL, PostScript, AFP, EPL, ZPL, TEC, PDF, QMF, and RTF.
- Wide support for distribution channels: print, email, fax, and web
- Integration for major archival systems (that is, Documentum, FileNet, and so on)
- Integration with key ERP systems – (that is, SAP, JD Edwards, Scala)
- Native IBM i implementation

InfoPrint Designer – what does it do

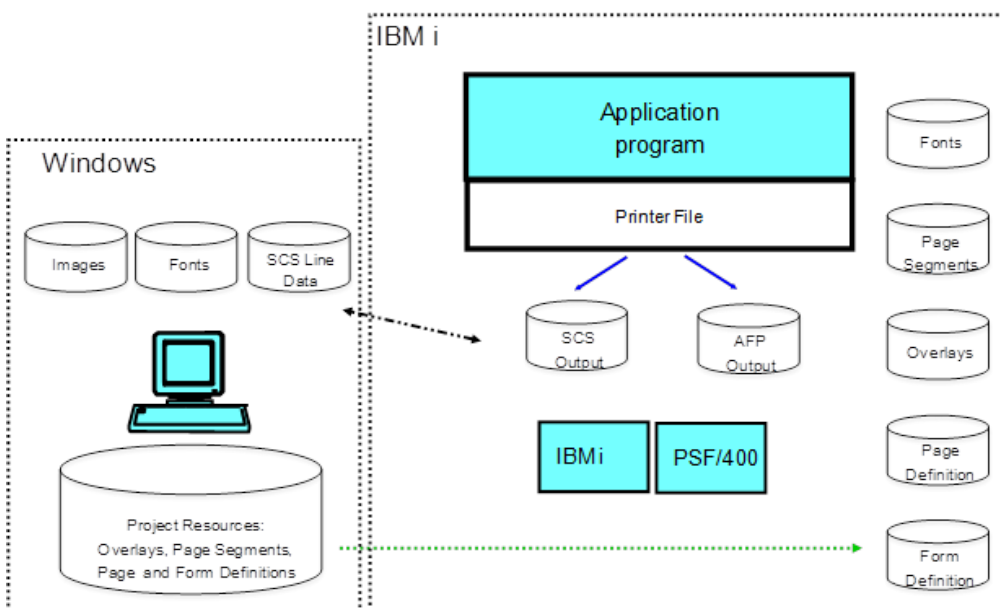
InfoPrint Designer (product number 5733–ID1) is an IBM i licensed program product for re-engineering existing line mode SNA character string (SCS) output into full page mode AFP. It was first released in 2001 with Version 5 Release 1 of the operating system, and consisted of the following three components:

- **Overlay Editor** for design of overlays (electronic forms)
- **Image Editor** for design of images (IBM i page segments)
- **Layout Editor** for actual design and placement of data and placement of other page elements (overlays, images, and barcodes)

InfoPrint Designer is Windows-based but does use IBM i Access for Windows to retrieve a target SCS spooled file from an output queue to use in the layout design and also for the upload of all designed component resources to the IBM i platform for production.

Using Figure 1, you can see how an application program on IBM i delivers output using the printer file definition to an SCS spooled file. The SCS spooled file is transferred to the InfoPrint Designer Windows platform to use as sample data for the design process. InfoPrint Designer applications are organized into projects that read in that sample data, along with the images and fonts to the design user interface.

Figure 1. InfoPrint Designer workflow



As you use the InfoPrint Designer graphical interface on Windows, in the background, InfoPrint Designer defines the format of the pages using the page definition and form definition resource

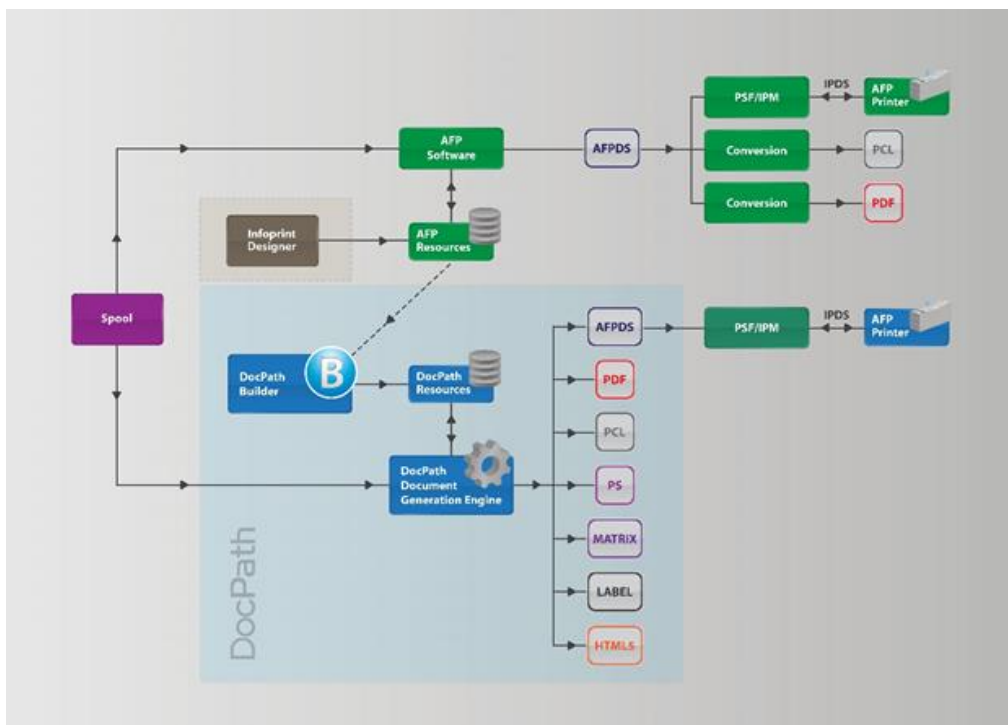
objects, which are elements of the AFP architecture. A sample SCS output file is downloaded for design. The completed resources, including images, overlays, and optionally fonts, are uploaded to IBM i. Putting the designed output into production involves only specifying the created page and form definition names in the application printer file. Then the OS will produce either AFP or PCL, depending on the options selected in the printer file.

While this process is well integrated on IBM i, there are problems, as we have covered. This is where DocPath Boulder Suite comes in handy. It addresses the conversion problem head-on, by providing a migration of existing InfoPrint Designer applications directly into DocPath format. This enables an InfoPrint Designer customer to migrate to the DocPath ecosystem seamlessly, with the output produced essentially identical. Then, as time and resources allow, the advanced design functions of DocPath can be brought to bear on each output application in priority sequence, enabling the most current, dynamic, advanced, and multi-channel customer communications.

The following information in this article takes you through the migration process.

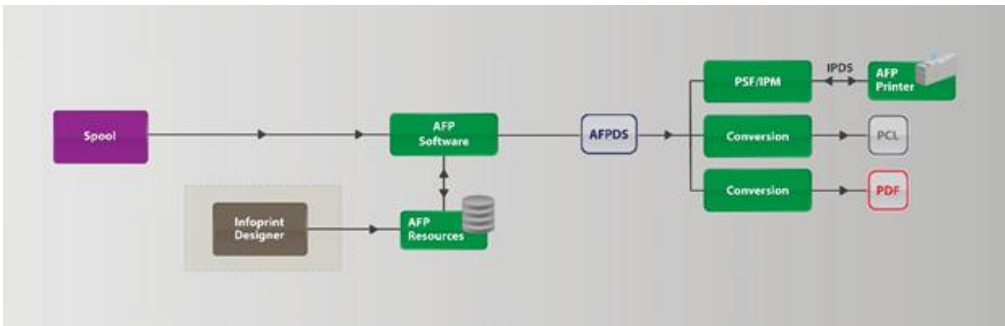
Introduction to DocPath Boulder Suite

Figure 2. Comparison of InfoPrint Designer and DocPath workflows



Boulder Suite is the DocPath suite of components that can be used to replace InfoPrint Designer projects. Figure 2 shows both: the InfoPrint Designer and the DocPath Boulder Suite workflows.

Figure 3. InfoPrint Designer schematic representation

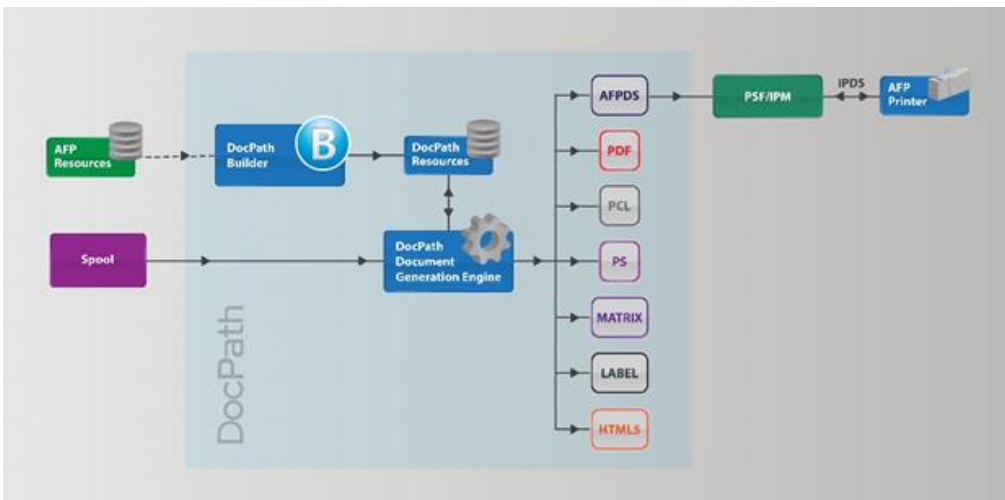


The InfoPrint Designer process (as shown in Figure 3) involves the following high-level steps:

- Spooled input is sent to be printed, but formatting is now guided by the page and form definitions specified in the printer file as designed in the InfoPrint Designer Windows interface.
- Output is to Advanced Function Presentation data stream to the output queue
- The AFP data stream can be printed to an Intelligent Presentation Data Stream (IPDS) printer using IBM Print Services Facility™ (PSF)
- The AFP data stream can also be converted to PCL using host print transform
- The AFP data stream can be converted to PDF using InfoPrint Server

When DocPath is put in place of InfoPrint Designer, the following workflow, as shown in Figure 4 takes place:

Figure 4. DocPath production workflow



- The original InfoPrint Designer project is converted to a DocPath Builder project.
- The project is placed on the integrated file system in the forms folder.
- Spooled input is sent to be printed.
- The input is processed by the DocPath generation engine.
- The DocPath generation engine used the project that was designed in DocPath Builder and placed in the forms folder.
- Multiple output formats can be generated simultaneously.

- The output can be printed, archived, and emailed under the control of the DocPath generation engine.

Boulder Suite consists of the following three separate components:

- The first one is DocPath Builder, which is a Windows .Net based application that provides the functionality to import the existing InfoPrint Designer projects, design form templates, and allows you to take full advantage of the Boulder Suite capabilities.
- The second component is the DocPath Document Generation Engine; a Java-based application responsible for creating the documents by merging the form templates with the data and applying the required logic. DocPath Document Generation Engine is a multi-tasking and scalable software application that ensures that you have the required processing power to respond to any increase in customer production.
- The third component is D-Forms, a fully native IBM i application that provides the user interface to set up, monitor, and communicate with the DocPath Document Generation Engine from the IBM i platform. All process definitions (rules) can be defined through D-Forms.

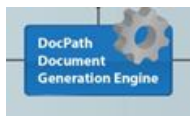
DocPath Builder overview



DocPath Builder is the design and conversion tool for Boulder Suite and is used to convert the InfoPrint Designer projects into the DocPath format. After the InfoPrint Designer project has been imported, DocPath Builder will show the data mapping, template design, and graphical flow (called JobFlow) of how the new migrated project will produce its output. Builder allows you to keep the current output format or change it to produce other output formats such as PDF and email.

After the project is complete, you compile the project and upload it to IBM i. A utility is provided with Builder to make the upload to IBM i a simple process.

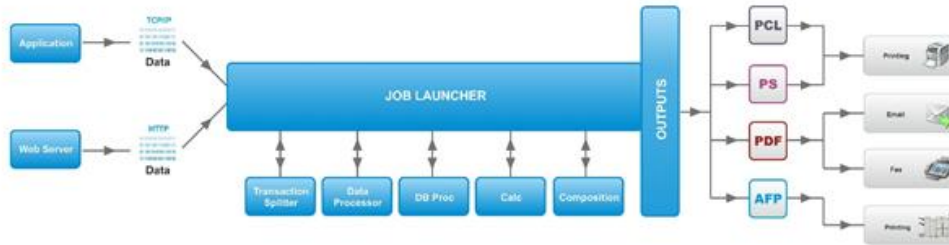
DocPath Document Generation Engine overview



The DocPath Document Generation Engine is the engine responsible for data processing and document generation, storage, and distribution. In order to ensure maximum performance on any production system, the engine has been entirely developed in Java. Java offers a wide range of advantages, especially to large enterprises. Being based on the client/server architecture, the Document Generation Engine offers great connectivity with large systems, interoperability, and flexible and efficient use of hardware resources. The engine operates both at an operating system level as well as at the Java Virtual Machine level. This means that DocPath solutions easily integrate simultaneously with legacy applications running on a specific OS, as well as with new Java-based programs.

The Document Generation Engine encompasses a wide range of DocPath modules and components that is described later in this article. Figure 5 shows how the Document Generation Engine resides in the application server environment and can communicate in various ways with different application platforms.

Figure 5. DocPath IBM i schematic



The Document Generation Engine is shown with a number of components controlled by the job launcher. The process runs inside a Java Application Server and accepts TCP/IP or HTTP-based data. This can be provided by an application, but in most cases, it is provided by the DocPath HTTP Client application that gets executed by D-Forms using the SENDTOJL (*send to job launcher*) command.

After a data file is sent to the job launcher in conjunction with the project to be used, the job launcher processes it using the components necessary to process the job based on the job flow in the project. The project data is extracted into the data schema and split into individual transactions. This enables concurrent processing of the data through the remaining processes. These include gathering additional information from a database, performing calculations, and generating the document composition. Multiple output formats can be generated at the same time. The output destination and the output type is also controlled by the job flow. It can be written to a printer, file, email or can be sent back to the calling application.

D-Forms overview



D-Forms is the automation tool for Boulder Suite. It is a native tool on IBM i and provides a full set of commands to automate functions.

Figure 6. DocPath D-Forms main menu

```

DFORMS      D-Forms Main Menu      DPTOOLS      RES1515

Select one of the following:

Administration
  1. D-Forms Setup
  2. Maintain distribution lists

Interactive
  10. D-Forms for iSeries          20. Start D-Output Manager Subsystem
  11. Work with Spool files       21. End D-Output Manager Subsystem
  12. Work with OneWorld Jobs    22. D-Output Manager Jobs

Production
  30. Define D-Output Manager Monitors  32. Work with Spool File Process Rules
  31. Work with log files             33. Work with OneWorld Process Rules

  90. Sign off
Selection or command                (C) COPYRIGHT DocPath, 1998-2014
==>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
F13=Information Assistant   F16=System main menu

```

D-Forms monitors output queues for certain spooled files based on user-defined criteria. When a spooled file that meets the required criteria is placed in a monitored out queue, a series of user-defined commands are run to call the Document Generation Engine as well as other IBM i commands to generate the required results. This collection of commands is referred to as a *rule*.

There can be up to three levels of selection criteria used for processing a spooled file:

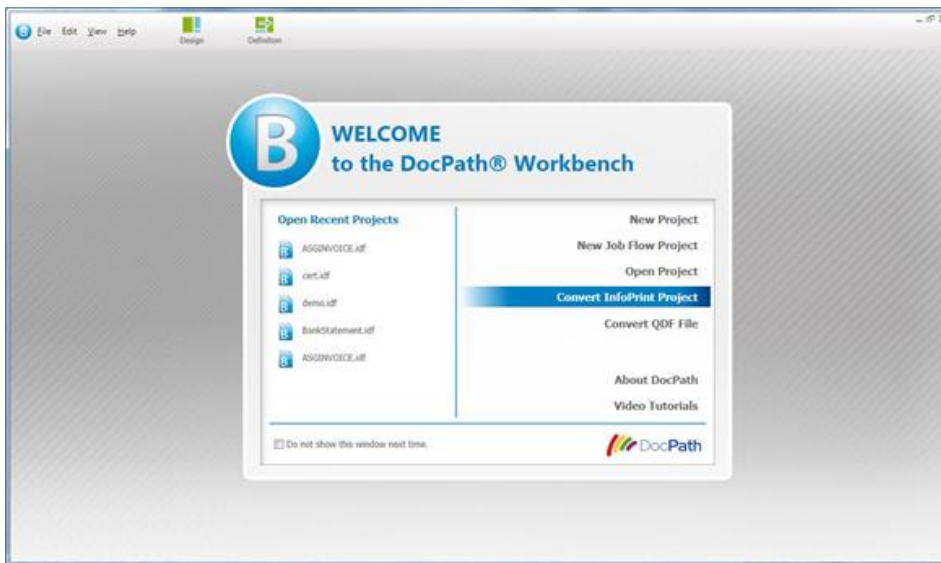
- Input file monitor Level
Spooled files are monitored based on one of the following three methods:
 1. Scheduled (time of day or other interval)
 2. Timer (every x number of seconds)
 3. Data Queue (spooled files are processed immediately)
Spooled files are then tested against the required file name, form type and user data information.
- Rule monitor level
Spooled files that meet the criteria of the monitor are then tested against the required criteria defined per rule and either linked to that rule name for processing or ignored.
- Command level
Spooled files linked to a rule name are processed by the specified command lists. Commands can be conditioned based on IF statements.

Typically, one command is required to interface with the Document Generation Engine. This is the SENDTOJL command and is used to send the contents of the spooled file to the Document Generation Engine. The Document Generation Engine will then process the spooled file and generate the output to the specified location.

This provides a description of the process and the components needed on the production side to produce the output. The next section describes how the objects are converted and put into production.

Converting an InfoPrint project and uploading

Figure 7. DocPath Workbench main menu



DocPath Builder has the ability to convert the formatting of InfoPrint Designer projects by translating the page and form definition objects into the DocPath Builder Project format (.idf). The process to convert a project is simple and you only need access to the files that belong to the project to convert them successfully.

In this example, the sample invoice that was supplied with InfoPrint Designer is converted.


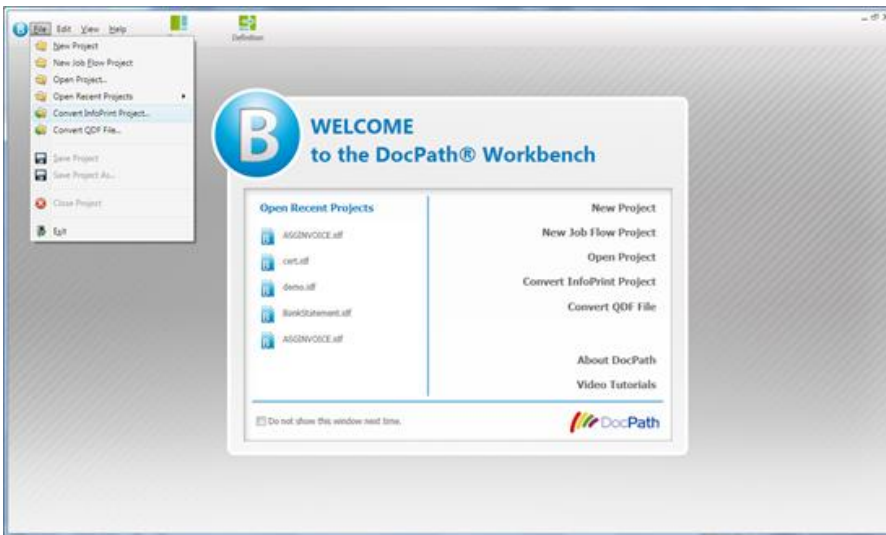
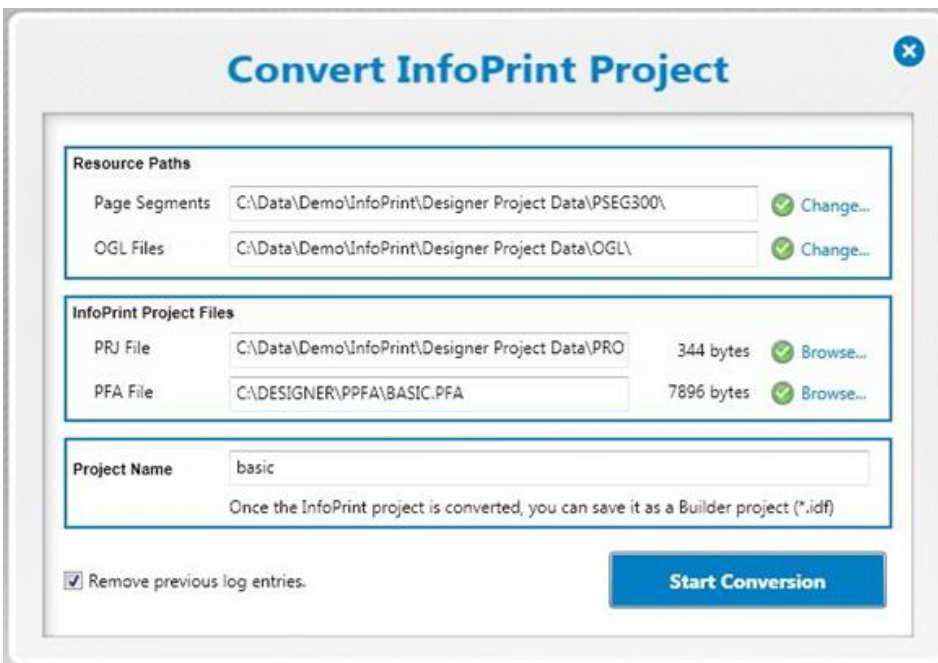
1. On the Welcome screen, select the project from the Open Recent Projects section to convert or click File  Convert InfoPrint Project.

Figure 8. Converting an InfoPrint project from the menu



The project conversion wizard is displayed.

Figure 9. Conversion directory paths



2. In the conversion wizard, specify the correct path for the page segments and Overlay Generation Language (OGL) files in the respective text fields.

Figure 10. Set resource paths



3. Then select the InfoPrint Designer files for the project and the page definition (PFA) formatting.

Figure 11. Select InfoPrint project files



4. After selecting these files, you are ready to name the project. By default, the name of the original InfoPrint Designer project is displayed in the Project Name field, but you can change it to any name of your choice. This will be the name of the DocPath Builder project with a .idf extension.

Figure 12. Name the project



5. Click **Start Conversation** to convert the project.

After the project is converted, you will be returned to the Builder home page. The name of the converted project will be to the right of the Definition icon and will be the name you selected for the project.

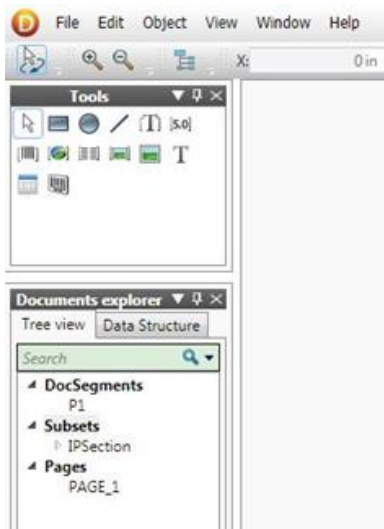
Figure 13. Builder shows the converted project components



All of the other option, such as **Pagination** will also be shown as the conversion also creates them as default for all of the other areas in the project so that it is almost ready to run.

6. Click **Design** to open the DocPath Designer and form here, you can review and change the converted project, if required.

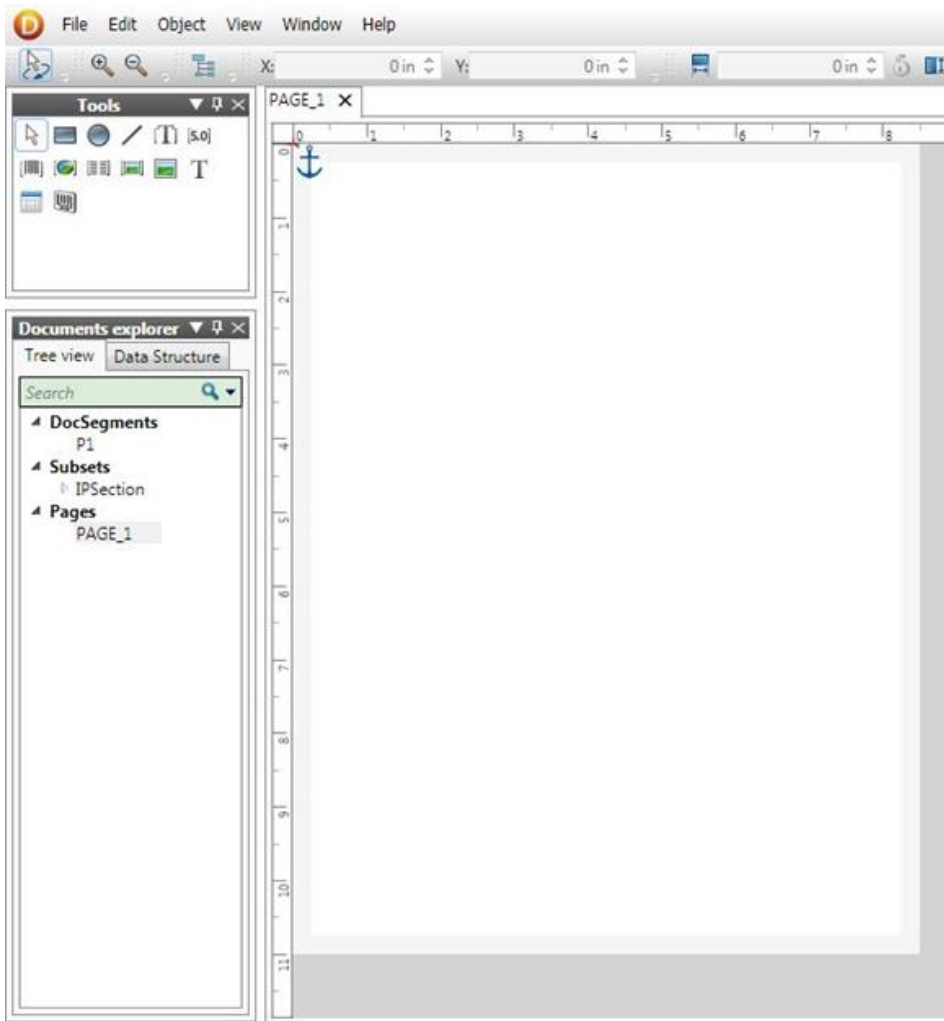
Figure 14. DocPath Designer user interface



The converted project template has been created and it consists of three design objects, a page called PAGE_1, a document segment called P1, and a subset which is a logical group of DocSegments called IPSection.

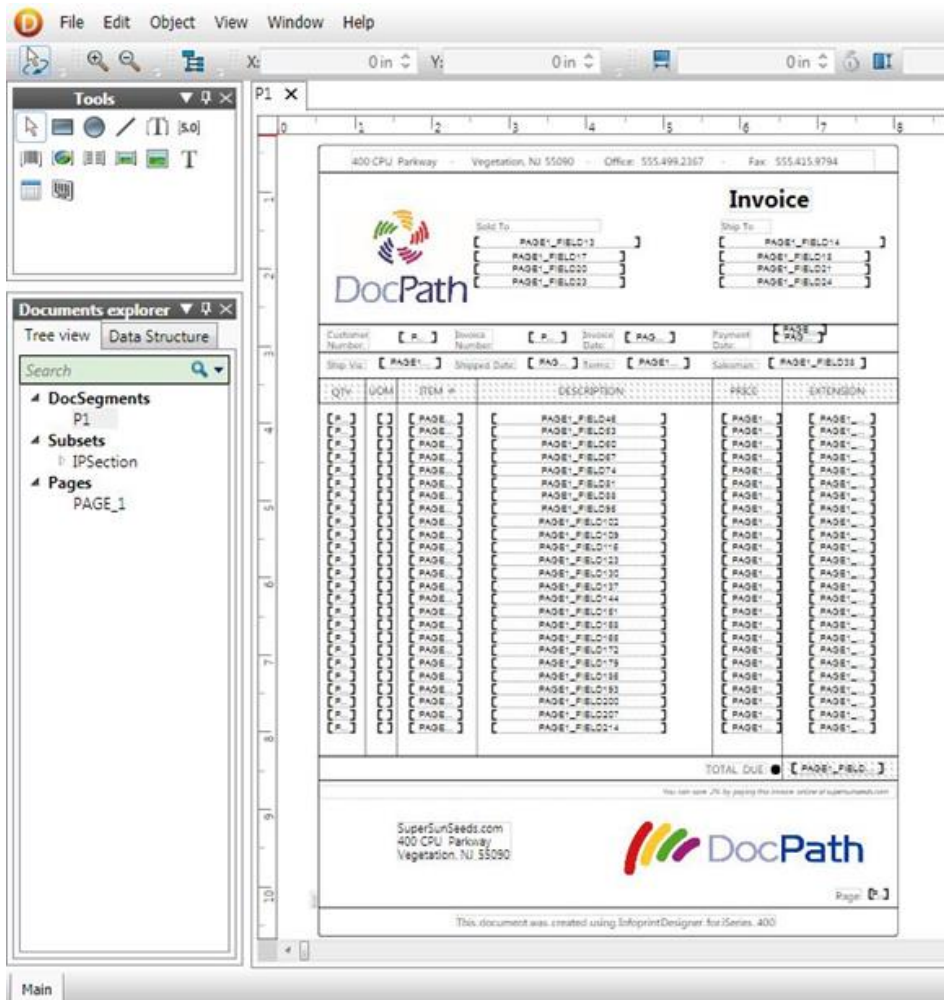
7. PAGE_1 is a blank page with an anchor at the top of the page. Due to all the real page layouts being converted to DocSegments, the anchor shows where the DocSegment will appear. This page is supposed to be blank as it acts like a blank canvas for the dynamic pieces of the form to be painted on it.

Figure 15. DocPath Designer page anchoring



8. The P1 DocSegment contains all data for the template as this is a one page form. All the fields are brought over in the conversion and you can easily modify the look and feel of the template to include updated logos or modify the fonts.

Figure 16. Migrated invoice example



9. After the design of the template is complete, you can use the graphical JobFlow interface to set up the type of output that will be produced and the destination of the output. By default, the job flow points to the AFP data stream and prints output. The job flow is a graphical representation of what happens within the project. It specifies the type of input, how the output will be processed, and the types of output to be produced along with the destination of that output. You can also have some logic so that the output flows in different directions depending on triggers.

Figure 17. Graphical job flow

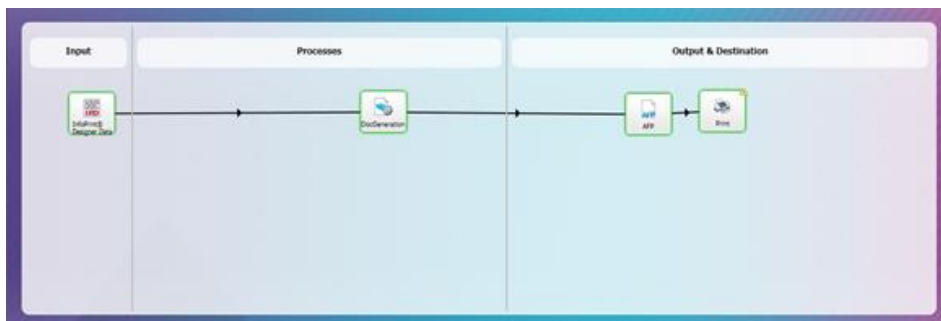
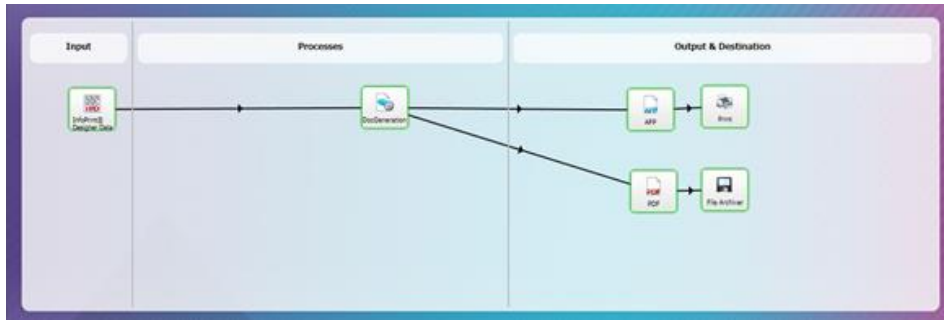


Figure 17 illustrates the simplest job flow workflow. The output and the destination printer is defined and the job would be ready to put into production.

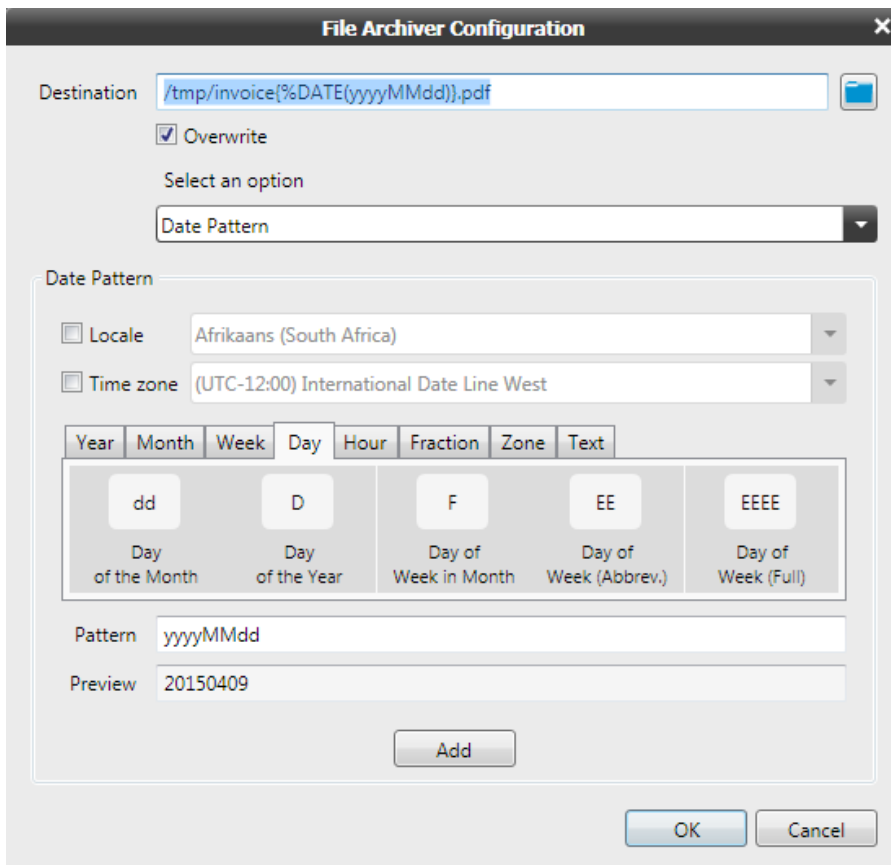
You can, however, enhance this workflow in many ways. Figure 18 shows the addition of an additional output to the PDF, along with the output "channel" of the integrated file system.

Figure 18. Job flow addition to output PDF



- 10. Setting up the name of the output PDF is done by choosing the path and the required name. This is done by using the icon's properties as shown in Figure 19. The name can consist of variables such as data, date, and time as well as counters. In this case, double-click the file archiver and choose the path and the base name, Invoice.pdf. In this example, I have also included the date, and therefore, the final name is **/tmp/invoice{%DATE(yyyyMMdd)}.pdf**.

Figure 19. Job Flow icon properties



- Double-click the InfoPrint Designer Data icon to verify the definition, though this does not usually need to be touched. In the definer the conversion wizard has automatically mapped all the fields that were in the original design.

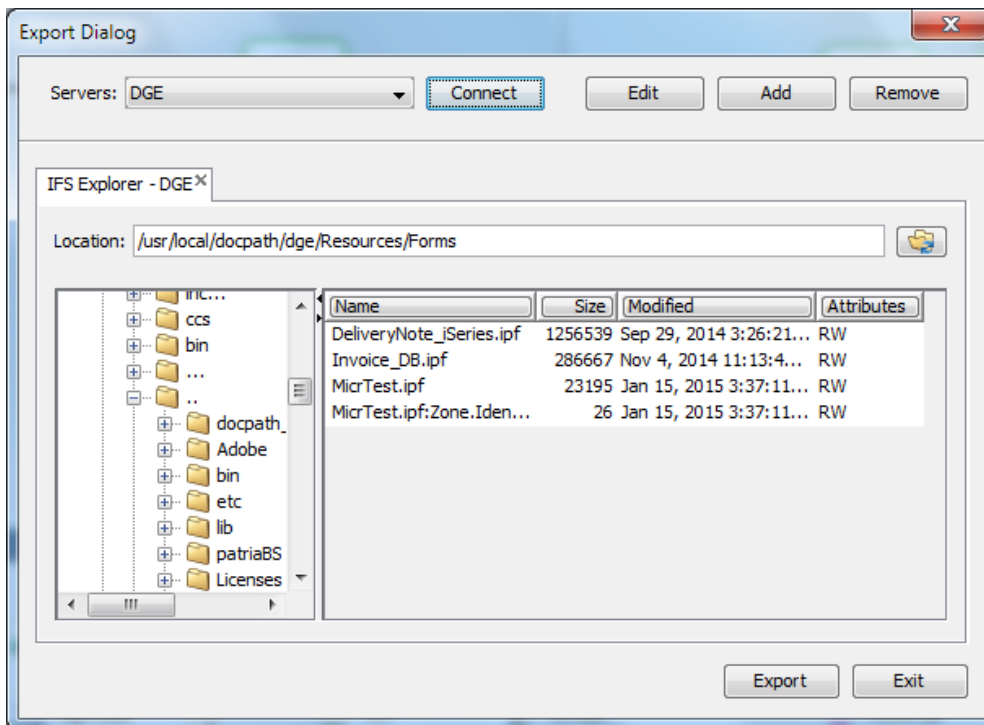
The Project is now ready to be compiled and sent to IBM i for testing.

Deploying the project to IBM i

Projects can be deployed in three ways to IBM i. The standard method is to use the **Compile and Send** option. Alternatively, you can copy the file directly to the integrated file system or use FTP.

The **Compile and Send** option is accessed from the **File** menu in DocPath Builder. When you click **Compile and Send**, a .ipf file is created and then using the connection dialog, you can log in to IBM i and then export the compiled project (refer to Figure 20).

Figure 20. Export Dialog



After exporting the document, it appears in the list and then you can close the dialog by clicking Exit.

DocPath Boulder Suite automation

Using the Document Generation Engine and D-Forms on IBM i

The Document Generation Engine makes use of the IBM i native component called D-Forms to provide native IBM i front-end and the connection to the Java-based Document Generation Engine components.

D-Forms allows you to use the IBM i green screen to set up monitors to monitor the output queues and then run rules against the spooled file output in those queues that meet certain criteria.

D-Forms will also allow you to run native IBM i commands and a number of other functions that are included, all designed to make integration easier.

The next step is to set up a monitor and define a rule that can send data to the Document Generation Engine and tell it what project will be used to be merged with the data to produce the necessary output.

Starting the Document Generation Engine

When you install Boulder Suite, the Document Generation Engine gets installed in its own instance called DPTools.

The DPTools instance has its own subsystem that needs to be started. When the subsystem is started, the Document Generation Engine is started with it.

Create the rule and monitor

Using Option 30 from the D-Forms main menu, you can define an outqueue monitor to monitor a single outqueue or multiple outqueues.

1. In option 30, press F6 to create a new monitor and enter a name for the monitor.

Figure 21. Monitor definition



```
DAMMNT      Add D-Output Manager Monitor definition
Type Rule, press Enter.
Monitor . . . INVOICEPRT
```

2. Specify the outqueue that will be used. You can also monitor multiple outqueues, if required.

Figure 22. D-Output monitor definition

```
DAMMNT      Add D-Output Manager Monitor definition
Enter details and press Enter.

Monitor    . . : INVOICEPRT

Outq name  . . : INVPRT      Outq name, *MULTIPLE, *FILE
Library    . . : QUSRSYS

Monitor type . *DTAQ_  *DTAQ, *TIMER, *SCHED

F3=Exit
Outq does not exist. Press Enter to create.
```

If an outqueue does not exist, you will be prompted to create it. The same is true if you are using a data queue.

3. There are other options that can be set for the monitor, but in most cases, you can use the default values.

Figure 23. Add D-Output Manager Monitor

```
OAMMNT          Add D-Output Manager Monitor definition
Enter details and press Enter.

Monitor . . . : INVOICEPRT

Outq name . . : INVPRT
Library . . . : DFORMSQA

Monitor type : *DTAQ
Dataqueue . . : INVPRT
Library . . . : DFORMSQA

User name . . : *ANY + User, *ANY
Form type . . : *ANY Type, *ANY
User data . . : *ANY User data, *ANY

Job desc. . . : OAMJOB +
Library . . . : DFORMSQA

Write log . . : Y Y=Yes, N=No

F3=Exit F4=Prompt F12=Previous
```

4. After the monitor is created, use option 8 to start it.

Figure 24. Start the monitor

```
WRKOAM          Work with D-Output Manager monitors          RESI515
Type options, press Enter
2=Edit 4=Delete 5=Display 8=Start 9=End

Opt  Monitor      Type      Status      Outq      User      Form Type  User Data
--  -
8  INVOICEPRT      *DTAQ     *STOP      INVPRT    *ANY      *ANY      *ANY
```

Commands to process the spooled file

Next, you need to set up the commands to automate the processing of the spooled file. To do this, use option 32 from the D-Forms main menu.

In this example, invoice spooled files that come in on the INVPRT outqueue and that have a form type of INV will be processed. By setting up the rules in this manner, you can control how spooled files get processed. In this case, the spooled file is being sent for Document Generation Engine processing so that only a single command is needed in the rule, and that is SENDTOJL (*send to job launcher*).

1. After selecting option 32 from the main menu, press F6 on the rules screen to add a new rule. All the options needed to specify to meet the desired processing criteria is displayed on the rule screen.

Figure 25. Add rules



2. Press Enter to display the list of rules. Use option 6 to enter the command.

Figure 26. Select the rule



3. Press F6 to enter a new command. Any IBM i command or one of the many utilities that are provided by D-Forms can be used. In this case, enter the SENDTOJL command.

Figure 27. Adding the SENDTOJL command



4. Fill in the prompts for the SENDTOJL command using some specific names such as the name of the project and then a number of variables that are available, such as NAM (Job Name), so that the job name for the spooled file will get substituted when the automation job is running.

Figure 28. Sending file to job launcher



After the command is configured, it will be shown in the list of commands that will be run when an incoming spooled file with the correct attributes triggers the rule to execute.

Figure 29. List of commands to be processed

```

PROCCMD      Commands to be processed      RESI515

Type options, press Enter                    Rule name . : INVOICEPRT
2=Edit 3=Copy 4=Delete 6=Edit command 8=Comment on/off

Opt Seq. No.  Command String
-----
 10.00 SENDTOJL DGELIB (DGE) IPFINP (BASIC.IPF) FILE (&SNAM) JOB (&JNUM/&US
ER/&JNAM) SPLNBR (&SNUM)
    
```

After the monitor is started, any spooled file coming from the INVPRRT outqueue with a form type of INV will get processed by this rule.

Testing the output

To test the output place a spooled file in the output queue with a ready status. This will run the automation process and the output will be produced based on the JobFlow specified in the project.

Testing the project design

There are two types of testing when you build a project.

1. First, you can preview the document in DocPath Builder making use of the sample data that you downloaded when you verified the data mapping in the definer using the InfoPrint Designer Data definition. This runs a local copy of the Document Generation Engine which will generate a temporary PDF document for you to view. This is the easiest way to verify the project design without running it on IBM i.
2. Second, you can test the design on IBM i using the SENDTOJL command interactively on IBM i. It will submit the job to the Document Generation Engine and write the result as specified in the job flow of the .ipf file. This is how you would perform manual testing before you create the automation steps for the job using the DFORMS rules.

Figure 30. Sample SENDTOJL command

```

SEND FILE TO JOB LAUNCHER (SENDTOJL)

Type choices, press Enter.

DGE Library Name . . . . . _____ Character value
IPF Project File . . . . . _____

-----
Type of Input . . . . . *SPLF          *SPLF, *DBF, *STMF
Spool File . . . . . _____         Name
Job Name . . . . . * _____         Name, *
 *USER . . . . . _____         Name
 Number . . . . . _____         000000-999999, ' '
Spool File Number . . . . . *ONLY       1-999999, *ONLY, *LAST
Range of Pages to Copy:
 Starting Page . . . . . *SPLFA         Number, *FIRST, *LAST, *SPLFA
 Ending Page . . . . . *SPLFA         Number, *LAST, *FIRST, *SPLFA
Type of Input . . . . . *FCFC          *FCFC, *NONE
Codepage of Input File . . . . . _____ Character value

More...
    
```

Note: You need to be in DFORMS so that your library list will be correct and the SENDTOJL command will be available.

Boulder Suite: Advanced considerations

DocPath Boulder Suite enables seamless migration of InfoPrint Designer applications. After an application is migrated, the decision can be made to take advantage of the advanced capabilities in Boulder Suite. Of course, new applications can also be designed and deployed using Boulder Suite. Refer to the second DocPath DeveloperWorks feature, *Using DocPath Advanced Functions*, for guidance.

This follow-on article will cover the following topics:

- Data mapper and base design interfaces
- Tables – fully dynamic tables and page overflow control
- Tables – conditional control of all elements of a table – rows, columns, cells
- Lists – data-driven lists with page overflow, sortation, and conditional control
- Charts – 2D and 3D dynamic charting (pie, bar, and so on) with control over all elements
- Object handling – anchoring, resizing, rotation
- Images – along with dynamic resizing
- Support for page layers – that is, different layers for different language versions
- Business rules – presentation based on conditions using wizard or scripting
- White space – marketing messages

Summary

In this article, the process to migrate InfoPrint Designer applications to DocPath Boulder Suite has been covered. Once migrated, applications can continue to function essentially how they have been functioning. Or, you can take advantage of the advanced DocPath platform and create complex, dynamic communications in all the *channels* that customers expect. In addition, you would of course also use DocPath to develop new document and customer communication applications.

Resources

- Learn about [DocPath Boulder Suite](#).
- Refer to the [DocPath Document Management solution video](#).
- Refer to [IBM InfoPrint Designer Migration & Replacement: DocPath Boulder Suite](#).
- Refer to [Convert IBM InfoPrint Designer projects into DocPath resources](#).
- Reach out to support@docpath.com for queries on technical information and sales@docpath.com for product-related information.

© Copyright IBM Corporation 2015
(www.ibm.com/legal/copytrade.shtml)

[Trademarks](#)

(www.ibm.com/developerworks/ibm/trademarks/)