

QMF Analytics for Multiplatforms
Version 12 Release 2

*Getting Started with QMF Analytics for
Multiplatforms*



Note

Before using this information and the product it supports, be sure to read the general information under "Notices" at the end of this information.

August 2, 2021 edition

This edition applies to IBM® QMF Analytics for Multiplatforms Version 12 Release 2, program number 5737-E42. The content in this edition applies to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1995, 2020.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

- Tables..... xi**

- About this information..... xiii**
 - Who should read this information..... xiii
 - Service updates and support information..... xiii
 - Highlighting conventions..... xiii
 - How to read syntax diagrams..... xiii
 - How to look up message explanations..... xiv
 - How to send your comments..... xv

- Chapter 1. QMF overview..... 1**
 - User perspective overview..... 1
 - Administrator perspective overview..... 1

- Chapter 2. Using the QMF interface..... 3**
 - Working with views and perspectives..... 6
 - Opening perspectives..... 6
 - Closing perspectives..... 6
 - Opening views..... 7
 - Positioning views..... 7
 - Configuring perspectives..... 7
 - Resetting perspectives..... 8
 - Setting preferences..... 8
 - Setting general preferences..... 8
 - Setting appearance preferences..... 9
 - Setting dashboard runtime preferences..... 9
 - Working with global variables..... 10
 - Setting help preferences..... 11
 - Setting JDBC preferences..... 13
 - Setting LOB preferences..... 14
 - Setting log preferences..... 15
 - Setting the search configuration..... 15
 - Setting SMTP server preferences..... 16
 - Setting the directory location preference for objects generated by Export actions..... 17
 - Setting root directory location for dynamart source files..... 17
 - Setting third party library preferences..... 18
 - Specifying Visual Designer preferences..... 18
- Exploring the Workspaces view..... 19
 - Discovering the contents of a workspace..... 19
 - Workspace menu commands..... 20
 - Working with QMF search..... 21
 - Managing workspace folders..... 22
 - Filtering data source objects..... 23
 - Creating links to workspace objects..... 23
 - Creating links to repository objects..... 23
 - Viewing properties for workspace objects..... 24
 - Refreshing workspace contents..... 25
 - Moving workspace objects..... 25
 - Adding content to a workspace..... 25
 - Creating additional workspaces..... 26

| | |
|-----------------------------------|----|
| Working in the Personal view..... | 26 |
| Personal view menu commands..... | 27 |

Chapter 3. Accessing the data..... 29

| | |
|---|----|
| Repositories and data sources..... | 29 |
| Creating personal repositories..... | 30 |
| Connecting to a repository..... | 30 |
| Specifying user information..... | 31 |
| Changing user interface configuration information..... | 31 |
| Setting up JDBC driver information..... | 32 |
| Setting up repository connection information..... | 34 |
| Configuring repository caching..... | 42 |
| Importing Settings from QMF for Windows..... | 45 |
| QMF for Windows settings..... | 45 |
| Correct format for identifiers..... | 46 |
| Matching patterns..... | 46 |
| Creating new objects or repository items..... | 46 |
| Creating web links to data objects..... | 48 |
| Saving objects..... | 49 |
| Saving objects to a file..... | 50 |
| Saving objects in a repository or a repository workspace..... | 50 |
| Saving objects in the QMF catalog..... | 51 |
| Opening saved objects..... | 51 |
| Opening QMF catalog objects..... | 52 |
| Opening repository objects..... | 53 |
| Opening objects saved to a file..... | 53 |

Chapter 4. Creating queries..... 55

| | |
|--|----|
| Query types..... | 55 |
| Query interfaces..... | 56 |
| Creating visual queries using the SQL editor..... | 58 |
| Creating visual queries using SQL..... | 58 |
| Using Content Assist for visual queries..... | 59 |
| Using Parameter Hints for visual queries..... | 60 |
| Creating visual queries using the Prompted Query editor..... | 60 |
| Adding tables..... | 62 |
| Joining tables..... | 63 |
| Joining columns..... | 64 |
| Specifying columns for query results..... | 64 |
| Specifying sort conditions for query results..... | 65 |
| Specifying row conditions for query results..... | 66 |
| Creating visual queries in the Query Builder editor..... | 67 |
| Adding and joining tables in the Query Builder editor..... | 68 |
| Editing join conditions for tables..... | 69 |
| Specifying fields and conditions for a visual query..... | 70 |
| Specifying global row conditions..... | 72 |
| Editing SQL text..... | 74 |
| Creating visual queries using the Draw Query wizard..... | 74 |
| Listing data source tables..... | 75 |
| Working with prompts in queries..... | 76 |
| Working with SQL prompts..... | 77 |
| Working with simple prompts..... | 77 |
| Working with prompt hierarchies..... | 80 |
| Working with shared prompts..... | 84 |
| Creating OLAP queries..... | 86 |
| Retrieving cube data..... | 87 |
| Opening existing OLAP queries..... | 88 |

| | |
|--|------------|
| Modifying OLAP queries to obtain different views of cube data..... | 88 |
| Analytical queries and associated transformations..... | 91 |
| Building analytical query structures..... | 92 |
| Populating analytical query transformations..... | 93 |
| Configuring analytical query transformations..... | 95 |
| Profiling queries..... | 115 |
| Publishing objects to QMF Vision..... | 116 |
| Working with database tables..... | 117 |
| Specifying sort conditions for tables..... | 121 |
| Specifying row conditions for tables..... | 123 |
| Editing database tables using Db2 special registers..... | 124 |
| Copying tables..... | 125 |
| Chapter 5. Selecting a new data source..... | 127 |
| Chapter 6. Setting fonts for SQL statements..... | 129 |
| Chapter 7. Specifying substitution variables..... | 131 |
| Chapter 8. Working with procedures..... | 133 |
| Creating a new procedure..... | 133 |
| Creating JavaScript procedures..... | 134 |
| Creating procedures as flowcharts..... | 135 |
| Editing flowchart procedures..... | 136 |
| Adding JavaScript conditions to procedures..... | 136 |
| Procedure commands..... | 137 |
| Interface differences..... | 137 |
| Procedure syntax..... | 137 |
| Reading the syntax diagrams..... | 138 |
| QMF Catalog objects versus Repository folder objects..... | 139 |
| BOTTOM command..... | 140 |
| CONNECT command..... | 141 |
| CONVERT command..... | 141 |
| DISPLAY command..... | 142 |
| DRAW command..... | 143 |
| EDIT command..... | 144 |
| ERASE command..... | 144 |
| EXECUTE command..... | 145 |
| EXPORT command..... | 145 |
| FORWARD command..... | 150 |
| IMPORT command..... | 150 |
| LIMIT LOCAL command..... | 151 |
| MAIL TO Command..... | 152 |
| PRINT command..... | 155 |
| RESET GLOBAL..... | 158 |
| RUN command..... | 158 |
| RUNTSO command..... | 161 |
| SAVE command..... | 164 |
| SET GLOBAL command..... | 166 |
| SET INVISIBLE command..... | 167 |
| SET LOCAL command..... | 167 |
| SET LOCAL WITH VALUES command..... | 168 |
| SET OPTIONS command..... | 168 |
| SHOW command..... | 169 |
| USE REPOSITORY command..... | 170 |
| JavaScript functions in procedures..... | 171 |
| The proc.exec () function..... | 171 |

| | |
|--|-----|
| The proc.exists() function..... | 172 |
| The proc.getVariable() function..... | 172 |
| The proc.getCellValue() function..... | 173 |
| The proc.getColumnname() function..... | 173 |
| The proc.getColumnType() function..... | 174 |
| The proc.getNumColumns() function..... | 174 |
| The proc.getNumRows() function..... | 174 |
| The proc.include() function..... | 175 |
| The proc.messageBox() function..... | 175 |
| The proc.prompt() function..... | 176 |
| The proc.setCellValue() function..... | 177 |

Chapter 9. Managing query results..... 179

| | |
|---|-----|
| Settings for query results..... | 179 |
| Setting Row Limit..... | 179 |
| Setting Resource Limit..... | 179 |
| Formatting the query results..... | 180 |
| Using the Properties view..... | 180 |
| Using the Layout Properties dialog box..... | 180 |
| Filtering query results..... | 187 |
| Working with calculated columns..... | 190 |
| Adding calculated columns to a query result set..... | 190 |
| Adding custom functions..... | 191 |
| Applying categorization to query results..... | 193 |
| Adding conformance rules for categorization..... | 194 |
| Adding expressions for categorization..... | 194 |
| Applying grouping and aggregation to query result columns..... | 195 |
| Grouping and aggregation fields..... | 195 |
| Exporting query results..... | 196 |
| Sending query results in an email..... | 197 |
| Exporting query results to a database..... | 198 |
| Exporting query results to a file..... | 200 |
| Importing query results from CSV and text files..... | 203 |
| Specifying separator options for CSV and TXT files..... | 204 |
| Specifying the result set structure of CSV and Text files..... | 205 |
| Importing data contained in IXF files..... | 206 |
| Opening the contents of an IXF file for further processing..... | 206 |
| Directly importing an IXF file into a database table..... | 207 |
| Importing query results from Excel files..... | 207 |
| Importing table contents of a saved file into a database table..... | 209 |
| Saving data objects as dynamarts..... | 210 |
| Managing source data files in dynamarts..... | 211 |
| Working with LOB data in the query results..... | 211 |
| Overriding LOB resource limit option..... | 212 |
| Viewing and editing the contents of a cell in the query results..... | 213 |
| Viewing multiple result sets..... | 214 |
| Results menu..... | 214 |
| Sorting query results..... | 216 |
| Specifying print options for query result sets and quick reports..... | 216 |
| Codes to insert specific data in a page header and footer..... | 217 |
| Saving data using the regular and fast save methods..... | 218 |
| Transforming the Results tab into the Results editing view..... | 219 |

Chapter 10. Working with scheduled tasks..... 221

| | |
|--|-----|
| Adding an execute object action..... | 223 |
| Specifying export query operation options..... | 226 |
| Specifying export query to mail operation options..... | 227 |

| | |
|--|-----|
| Specifying print query operation options..... | 229 |
| Specifying export quick report to mail operation options..... | 229 |
| Specifying export quick report operation options..... | 230 |
| Specifying export visual report to mail operation options..... | 231 |
| Specifying export visual report operation options..... | 232 |
| Adding a resource manipulation task..... | 233 |
| Specifying copy resource parameters..... | 234 |
| Specifying mail settings..... | 235 |
| Adding an action group..... | 236 |
| Adding actions to an action group..... | 237 |
| Removing actions from an action group..... | 237 |
| Setting iterator parameters..... | 238 |
| Managing prompts in an action group..... | 239 |
| Specifying variables for actions and action groups..... | 240 |
| Scheduling local tasks..... | 241 |
| Sending HTML tables in email..... | 242 |

Chapter 11. Generating classic and quick reports..... 245

| | |
|---|-----|
| Creating a new report..... | 245 |
| Selecting a classic report's data source object..... | 246 |
| Listing data source objects..... | 247 |
| Exporting reports to a file..... | 247 |
| Creating classic report forms..... | 248 |
| Designing a classic report form..... | 248 |
| Using form variables..... | 260 |
| Using usage codes..... | 261 |
| Saving classic reports..... | 263 |
| Saving a form with an associated data source object..... | 264 |
| Saving a form without an associated data source object..... | 264 |
| Working with quick reports..... | 264 |
| Creating quick reports..... | 265 |
| Editing quick reports..... | 282 |
| Modifying your quick report design..... | 284 |
| Form menu..... | 284 |

Chapter 12. Working with forecasts.....287

| | |
|--|-----|
| Creating forecasts..... | 287 |
| Specifying forecast data source options..... | 289 |
| Specifying forecast date parameters..... | 291 |
| Specifying forecast grouping hierarchy options..... | 292 |
| Specifying forecast strategy options..... | 294 |
| Specifying forecasting models options..... | 295 |
| Setting simple moving average parameters..... | 297 |
| Setting weighted moving average parameters..... | 297 |
| Setting single exponential smoothing parameters..... | 297 |
| Setting double exponential smoothing parameters..... | 298 |
| Setting Holt-Winters method parameters..... | 298 |
| Setting multiplicative decomposition parameters..... | 299 |
| Setting curve fitting parameters..... | 299 |
| Setting polynomial regression parameters..... | 300 |
| Setting neural network parameters..... | 300 |
| Monitoring forecast model performance..... | 301 |
| Observing trends..... | 302 |
| Observing seasonality..... | 304 |
| Observing cyclicity..... | 305 |
| Editing chart properties..... | 306 |
| Editing grid properties..... | 307 |

| | |
|---|------------|
| Exporting forecasts..... | 309 |
| Specifying print options for a forecast..... | 310 |
| Transforming the Results tab into the Results editing view..... | 311 |
| Chapter 13. Working with charts..... | 313 |
| Creating charts..... | 313 |
| Specifying chart type options..... | 314 |
| Specifying data series options..... | 314 |
| Specifying format options | 315 |
| Creating charts from a query results grid..... | 322 |
| Working with charts in the Display editor..... | 323 |
| Creating charts in the Display editor..... | 323 |
| Managing display modes..... | 324 |
| Editing query result set columns..... | 324 |
| Editing display mode layout structures..... | 325 |
| Editing charts..... | 326 |
| Exporting charts..... | 327 |
| Specifying print options for charts..... | 328 |
| Chapter 14. Working with drill-down paths..... | 329 |
| About drill-down paths..... | 329 |
| Creating drill-down paths..... | 329 |
| Managing drill-down path recording display modes..... | 331 |
| Specifying print options for a drill-down path..... | 332 |
| Chapter 15. Working with visual projects..... | 335 |
| Visual project templates..... | 335 |
| Creating template categories for visual projects..... | 336 |
| Creating visual reports..... | 337 |
| Creating reports using the visual report wizard..... | 338 |
| Opening a legacy visual report..... | 340 |
| Specifying data source connections for visual projects..... | 341 |
| Specifying queries for visual projects..... | 342 |
| Designing the main page of a visual report..... | 343 |
| Designing a fixed page for a visual report..... | 343 |
| Managing report data retrieval..... | 344 |
| Exporting reports to a file..... | 346 |
| Working with Ad hoc reports..... | 347 |
| Working with visual dashboards..... | 348 |
| Planning visual dashboards..... | 350 |
| Creating Visual Dashboards using the Visual Dashboard wizard..... | 351 |
| Importing a legacy Visionary world..... | 353 |
| Specifying data source connections for visual dashboards..... | 353 |
| Working with scenes..... | 354 |
| Working with visual dashboards in version control systems..... | 393 |
| Displaying data on a map according to address information retrieved from a query..... | 394 |
| Protecting visual project data..... | 395 |
| Data source connections and visual projects..... | 395 |
| Queries and visual projects..... | 396 |
| Specifying queries for visual projects..... | 396 |
| Adding custom JavaScript functions to visual projects..... | 397 |
| Creating JavaScript Modules..... | 398 |
| Working with the Visual Designer..... | 399 |
| Working with the Project Explorer..... | 399 |
| Working with the editor..... | 401 |
| Editor controls in the visual designer..... | 401 |
| Working with the Properties view..... | 402 |

| | |
|---|------------|
| Working with the Events view..... | 407 |
| Working with the Palette view..... | 408 |
| Working with the Layers view..... | 477 |
| Working with global resources..... | 477 |
| Profiling visual dashboards..... | 491 |
| Analyzing the complexity of a scene..... | 491 |
| Profiling a visual dashboard..... | 492 |
| Debugging visual dashboards..... | 493 |
| Editing queries of particular objects in visual projects..... | 494 |
| Chapter 16. Working with JavaScript Tables..... | 495 |
| Creating JavaScript Tables..... | 495 |
| Writing scripts to populate JavaScript Table..... | 496 |
| JavaScript Table Example..... | 497 |
| Editing JavaScript Tables..... | 499 |
| Chapter 17. Creating virtual tables..... | 501 |
| Specifying structures of virtual tables..... | 501 |
| Specifying cache expiration for virtual tables..... | 502 |
| Invalidating virtual table and JavaScript table cache manually..... | 503 |
| Chapter 18. Working with global variables..... | 505 |
| Specifying user global variables..... | 505 |
| Editing system global variable values..... | 506 |
| System defined global variables..... | 506 |
| DSQQW global variables..... | 506 |
| DSQAO global variables..... | 511 |
| DSQEC global variables..... | 513 |
| DSQDC global variables..... | 515 |
| DSQCP global variables..... | 515 |
| Chapter 19. Working with edit codes..... | 517 |
| Edit codes for character data..... | 517 |
| Edit codes for date data..... | 518 |
| Edit codes for graphic data..... | 518 |
| Edit codes for numeric data..... | 519 |
| Edit codes for time data..... | 521 |
| Edit codes for timestamp data..... | 521 |
| User defined edit codes..... | 522 |
| Implementing custom edit codes for classic reports..... | 523 |
| Edit codes for metadata..... | 527 |
| Chapter 20. Working with Analytics objects..... | 529 |
| Opening Analytics objects from QMF Catalog..... | 529 |
| Opening Analytics objects with a source..... | 529 |
| Opening Analytics objects without a source..... | 530 |
| Using procedure commands with Analytics objects..... | 530 |
| Exporting Analytics objects..... | 531 |
| Chapter 21. Customizing the application environment..... | 533 |
| Using the command line..... | 533 |
| List of command line parameters..... | 534 |
| Using the command bar..... | 535 |
| Specifying default print options..... | 536 |
| Viewing resource limits..... | 536 |
| About resource limits..... | 537 |

| | |
|---|------------|
| Specifying Timeouts resource limits..... | 537 |
| Specifying Limits resource limits..... | 538 |
| Specifying SQL Verbs resource limits..... | 538 |
| Specifying Options resource limits..... | 539 |
| Specifying Save Data resource limits..... | 541 |
| Specifying Binding resource limits..... | 541 |
| Specifying Object Tracking resource limits..... | 542 |
| Specifying LOB Options resource limits..... | 543 |
| Specifying Cache resource limits..... | 544 |
| Specifying Object List resource limits..... | 544 |
| Appendix A. Accessibility..... | 545 |
| Standard keyboard equivalents in QMF for Workstation..... | 545 |
| Standard keyboard navigation..... | 545 |
| Operating system accessibility..... | 545 |
| Assistive technology products..... | 546 |
| Navigation in QMF for Workstation..... | 546 |
| Navigation in QMF for WebSphere..... | 548 |
| Appendix B. Troubleshooting..... | 549 |
| Problem displaying images in QMF for WebSphere using Internet Explorer 8..... | 549 |
| Creating a response file..... | 549 |
| QMF for Workstation Win32 API reverts to previous version of QMF for Windows..... | 550 |
| Appendix C. Messages..... | 551 |
| Notices..... | 621 |
| Trademarks..... | 622 |
| Terms and conditions for product documentation..... | 622 |
| Privacy policy considerations..... | 623 |
| Glossary..... | 625 |
| Index..... | 633 |

Tables

| | |
|--|-----|
| 1. User perspective: Views..... | 3 |
| 2. Administrator perspective: Views..... | 4 |
| 3. QMF Classic perspective: Views..... | 5 |
| 4. Visual Designer perspective: Views..... | 5 |
| 5. The settings that you can import from QMF for Windows | 45 |
| 6. Columns and available actions..... | 70 |
| 7. Prompt types..... | 76 |
| 8. Original result set data..... | 105 |
| 9. Crosstab result set data..... | 105 |
| 10. Original result set data..... | 107 |
| 11. Normalized query results..... | 107 |
| 12. Alternate normalized query results..... | 108 |
| 13. Crosstabbed query results for furniture sales..... | 108 |
| 14. Normalized query results for furniture sales..... | 108 |
| 15. Original data..... | 110 |
| 16. Conditionally grouped data..... | 111 |
| 17. Parameters for running objects..... | 159 |
| 18. Parameters for saving query results to a database..... | 160 |
| 19. Language-to-language identifier..... | 163 |
| 20. Text options..... | 201 |
| 21. Query result set and quick report print options..... | 217 |
| 22. Codes to insert specific data..... | 217 |
| 23. Main section elements for a classic report..... | 249 |

| | |
|--|-----|
| 24. Breaks section elements for a classic report..... | 250 |
| 25. HTML variables..... | 252 |
| 26. Calculations option elements for a classic report..... | 253 |
| 27. Columns option elements for a classic report..... | 254 |
| 28. Conditions option elements for a classic report..... | 255 |
| 29. Detail section elements for a classic report..... | 255 |
| 30. Options section elements for a classic report..... | 257 |
| 31. Final section elements for a classic report..... | 258 |
| 32. Page section elements for a classic report..... | 259 |
| 33. Condition parameters for the Delete operation..... | 375 |
| 34. Buttons and their indexes..... | 376 |
| 35. Toolbar options..... | 435 |
| 36. Command line parameters..... | 534 |

About this information

This information describes how to use the QMF for Workstation and QMF for WebSphere® applications.¹ Always check the DB2® and IMS Tools Library page for the most current version of this publication: www.ibm.com/software/data/db2imstools/db2tools-library.html

Who should read this information

This information is intended for all QMF for Workstation and QMF for WebSphere users.

Service updates and support information

To find service updates and support information, including software fix packs, PTFs, Frequently Asked Questions (FAQs), technical notes, troubleshooting information, and downloads, refer to the following web page:

http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_Tools_for_z~OS

Highlighting conventions

This information uses the following highlighting conventions:

- **Boldface** type indicates commands or user interface controls such as names of fields, folders, icons, or menu choices.
- Monospace type indicates examples of text that you enter exactly as shown.
- *Italic* indicates the titles of other publications or emphasis on significant terms. It is also used to indicate variables that you should replace with a value.

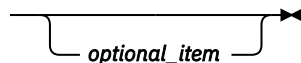
How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
 - The >>--- symbol indicates the beginning of a syntax diagram.
 - The ---> symbol indicates that the syntax diagram is continued on the next line.
 - The >--- symbol indicates that a syntax diagram is continued from the previous line.
 - The --->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).

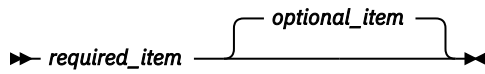
▶▶ *required_item* ▶▶

- Optional items appear below the main path.

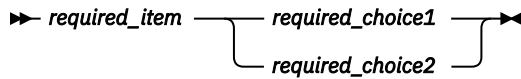
▶▶ *required_item* 

If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

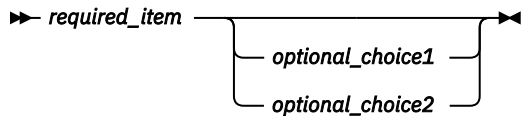
¹ Throughout this information, the IBM QMF client for WebSphere Application Server is referred to as QMF for WebSphere and the IBM QMF client for Workstation environment is referred to as QMF for Workstation.



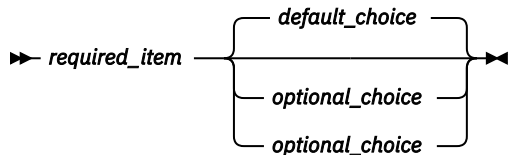
- If you can choose from two or more items, they appear vertically, in a stack. If you *must* choose one of the items, one item of the stack appears on the main path.



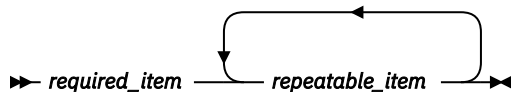
If choosing one of the items is optional, the entire stack appears below the main path.



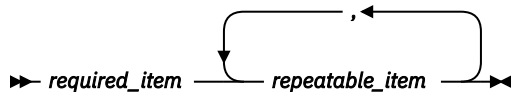
If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



- An arrow returning to the left, above the main line, indicates an item that can be repeated.



If the repeat arrow contains a comma, you must separate repeated items with a comma.



A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Keywords, and their minimum abbreviations if applicable, appear in upper case. They must be spelled exactly as shown. Variables appear in all lowercase italic letters (for example, *column-name*). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses; for example, (1).

How to look up message explanations

You can use any of the following methods to search for messages and codes.

Searching an information center

Use the following tips to help you improve your message searches:

- You can search for information on codes by entering the code; for example, enter -327.
- Enter the complete or partial message number. You can use wild cards in the message number to broaden your search; use * to represent multiple characters and use ? to represent any single character. For example:
 - The search string DFS20?I returns any messages that begin with the string DFS20, followed by any single character, followed by I.
 - The search string DFS20??I returns any messages that begin with the string DFS20, followed by any two characters, followed by I.
 - The search string DFS20*I returns any messages that begin with the string DFS20, followed by any number and type of characters, followed by I.

The information center contains the latest message information for all of the information management products that are included in the information center.

Using a Web search engine

You can use any of the popular search engines that are available on the Web to search for message explanations. When you type the specific message number or code into the search engine, you will be presented with links to the message information in IBM information centers.

Using LookAt

LookAt is an online facility that you can use to look up explanations for most of the IBM messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM message explanations for z/VM®, VSE/ESA, and Clusters for AIX® and Linux®:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at <http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/>.
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

More information is available in the LOOKAT.ME files available during the download process.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other QMF documentation, use either of the following options:

- Use the online reader comment form, which is located at:
 - www.ibm.com/software/data/rcf/
- Send your comments by e-mail to comments@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of QMF, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

Chapter 1. QMF overview

QMF is an all-purpose, multi-platform, Business Intelligence tool.

QMF is an Eclipse-based² Business Intelligence tool that transforms business data into a visual information platform with visual data on-demand. Using the QMF feature set, users can create intuitive dashboards, queries, reports, and procedures that can be easily distributed across an entire enterprise.

- Delivers critical enterprise-wide business information across multiple user and database platforms.
- Supports dynamic and static SQL, massive result sets, and stored procedure results.
- Supports simultaneously drawing data from Db2, Informix®, Oracle, SQL Server and most other JDBC-compliant data sources, as well as multi-dimensional data sources that support XMLA connectivity.
- Reports, dashboards, and queries can be optionally embedded within custom or 3rd party web applications, portals or workstation applications.
- Supports multi-dimensional queries, dashboards, reports, and layouts, as well as enhanced interoperability with existing BI infrastructure.
- QMF for Workstation is an Eclipse-based, cross-platform, rich workstation application that can be run on Microsoft® Windows®.
- QMF for WebSphere delivers broad functionality to enterprise users through ordinary Web browsers.

User perspective overview

The **User** perspective of the application interface provides the views, menus and wizards that enable you to perform an extensive range of query and reporting functions.

About this task

To get started using QMF, you should become familiar with the **User** perspective, the user preferences that you can set, and how you can work with the data sources and objects that are available to you in the **Workspaces** view. At any time, you can begin querying data sources, creating objects, formatting query results, or creating reports and dashboards.

Note: The application interface configuration information consists of the JDBC driver and repository connection settings that are used to connect to shared repositories and data sources. Advanced users can make changes to the configuration information from the **User** perspective. Most users will never change the configuration information.

Administrator perspective overview

The **Administrator** perspective of the application interface provides the views, menus, and wizards that enable you to create, manage, and maintain repositories.

A QMF administrator is responsible for setting up your application interface's configuration. As a result, the application interface that you are using is installed preconfigured with access to one or more

² Eclipse is an open source community whose projects are focused on providing an extensible development platform and application frameworks for building software. Eclipse provides extensible tools and frameworks that span the software development lifecycle, including support for modeling, language development environments for Java™, C/C++, PHP and others, testing and performance, business intelligence, rich client applications and embedded development. A large, vibrant ecosystem of major technology vendors, innovative start-ups, universities and research institutions and individuals extend, complement and support the Eclipse Platform. The Eclipse Foundation is a not-for-profit, member supported corporation that hosts the Eclipse projects. Full details of Eclipse and the Eclipse Foundation are available at www.eclipse.org.

repository workspaces. The data sources and objects that you can access from the workspace are displayed in the **Workspaces** view.

Chapter 2. Using the QMF interface

Use Perspectives, Views and Editors to set up and work in the user and administrator environments.

Perspectives

QMF uses and employs the Eclipse-based concept of *Perspectives*, *Views*, and *Editors*.

The sections that follow explain *Perspectives*, *Views*, and *Editors* as used by QMF. For information about the Eclipse concept of Perspectives, Views and Editors consult the [Eclipse documentation](#).

A Perspective is a way to organize and view the files and artifacts used with QMF.

The Perspectives that come with QMF are configured with set of *default views*, that are docked to specific areas in the application interface.

Administrators and users can select the Perspectives, Views, and Editors that suit their roles. For example, if your primary role is to design and develop visual dashboards, you will likely perform much of your work in the **Visual Designer** perspective.

Additionally, administrators and users can customize the interface so that whenever they launch QMF, it opens to a specific perspective and set of views.

To see the Perspectives available for use with QMF, perform **Window > Open Perspective > Other** from the menu bar.

QMF for Workstation and QMF for WebSphere provide the following perspectives:

- **Administrator** which provides the views, menus, and wizards that enable a user with administrative privileges to create and maintain repositories. Not all instances of QMF are installed with the **Administrator** perspective as a selected feature.
- **QMF Classic** which provides QMF TSO users with the capability to switch to QMF for Workstation without any difficulties as its interface is similar to QMF TSO.
- **QMF** which provides the views, menus, and wizards that are most appropriate for working with QMF catalogs.
- **Scene Script** which provides the interface from which you can create and work with visual dashboards by using JavaScript dynamically.
- **User** which provides the views, menus, and wizards that you will use to query data sources and work with query results.
- **Visual Designer** which provides the views, menus, and wizards that you will use to create visual reports and dashboards.

Views

A View is typically used to navigate a hierarchy of information, open an editor, or display properties for the active editor.

QMF views support editors and provide alternative presentations as well as ways to navigate the information in your window. In Eclipse, the layout of views within a page is controlled by the active perspective. This means that each of the perspectives (User, Visual Designer, and Administrator) have a designated or predetermined set of *Views*. Each view displays as a pane in the main workstation window. Each perspective of QMF initially displays the views that are most commonly used to perform specific functions.

| Table 1. User perspective: Views | |
|----------------------------------|---|
| View Name | Purpose |
| Analyzer | The Analyzer view displays the weight count of each object lying on a scene. |

Table 1. User perspective: Views (continued)

| View Name | Purpose |
|--------------------------------|--|
| Bookmarks | The Bookmarks view manages bookmarks for drill-down paths. |
| Cheat Sheets | The Cheat Sheets view displays a series of tutorials that outline the steps required to implement several basic user and administrator functions. |
| Data Source Connections | The Data Source Connections view displays a list of all connections that are currently active or were created to a data source. |
| Help | The Help view launches the Help for the application. |
| Outline | The Outline view displays the structural elements of the object that is currently open in the editor area. |
| Output | The Output view displays informational and error messages generated while running the application interface. |
| Personal | The Personal view maintains and displays objects in the Recently Used , Favorites , and Startup folders. |
| Profiler | The Profiler view displays the data on query running, the amount of created objects, and the time spent on creating them and calculating expressions on the visual project. |
| Progress | The Progress view displays all progress, percentage of work completed, and cancellation messages. |
| Properties | The Properties view displays the properties of the object that is selected or active in the editor area or another view. |
| Query Profiler | The Query Profiler view displays the time spent on running a query and fetching its results. |
| Repository Connections | The Repository Connections view displays the list of repository connections that you have available. |
| Repository Explorer | The Repository Explorer view displays the contents of the repositories to which you have connected. It also displays information about the references one object has on other objects. |
| Results Navigator | The Results Navigator view allows switching between results of a query or drill-down path. |
| REXX console | The REXX console view supports work with REXX procedures. Restriction: This feature is not available in the 64-bit application. |
| Search | The Search view displays the results of a search. |
| Workspaces | The Workspaces view lists the workspaces that are available to your user ID. All of the data sources that you can access are contained in the workspaces that are presented. From the Workspaces view you can perform the majority of query and reporting functions. |

Table 2. Administrator perspective: Views

| View | Purpose |
|---------------------|---|
| Repositories | The Repositories view displays the repositories that you created. This view is only available if you have the Administrator component. |

Table 2. Administrator perspective: Views (continued)

| View | Purpose |
|-------------------------|---|
| Users and Groups | The Users and Groups view displays the users and groups created for your internal security provider. |

Table 3. QMF Classic perspective: Views

| View | Purpose |
|---------------------|---|
| QMF Commands | <p>The QMF Commands view displays a list of function keys for a variety of supported command operations. The following function keys are listed:</p> <ul style="list-style-type: none"> • Help launches the online help. • List launches the Object List dialog listing the Data Sources. • End closes the workbench. • Query displays the query editor. • Retrieve launches the Retrieve command prompt. • Edit Table launches the Table editor. • Form launches the Form editor. • Proc launches the Procedure editor. • Report launches the Report editor. |

Table 4. Visual Designer perspective: Views

| View | Purpose |
|-------------------------|--|
| Events | The Events view displays navigation elements that you can apply to layout objects within a visual project. |
| Expressions | The Expressions view is used for debugging visual dashboards. |
| Layers | The Layers view displays the groups of layers. |
| Palette | The Palette view displays the objects that you can insert in a visual project. |
| Project Explorer | The Project Explorer view displays each element in a visual report and dashboard. |
| Properties | The Properties view displays the properties of the object that is selected or active in the editor or another view. |
| QMF Calculator | The QMF Calculator is used for calculating expressions in visual projects. |

Editors

Depending on the type of file that is being edited, the appropriate editor is displayed in the editor area.

Primary functional differences between views and editors:

- There is generally only one instance of a given view per workbench page, but there can be several instances of the same type of editor.
- Editors can appear in only one region of the page, whereas views can be moved to any part of the page and minimized as fast views.
- Editors can be in a *dirty* state, meaning that their contents are unsaved and will be lost if the editor is closed without saving.

- Views have a local toolbar, whereas editors contribute buttons to the global toolbar.
- Editors can be associated with a file name or an extension, and this association can be changed by users.

Each of the QMF perspectives includes a single editor area. By default, the editor area opens in the center of the workstation window.

There are many editors in QMF that will open in the editor area based on what type of object you open or what type of function you perform. You will use many of these editors when you are writing queries, formatting query results, creating reports or creating visual dashboards.

Working with views and perspectives

The QMF query environment includes views and perspectives that can be customized to fit your needs.

The following sections describe how to:

- Open and close perspectives and views
- Position and dock views in the application interface
- Customize the application interface to open to a perspective and view that suits your role and responsibility.

Opening perspectives

You can open any perspective that is not currently open in the workstation window.

About this task

A perspective will stay open until you close that perspective. Each time QMF is started, all perspectives that were open at the time of the previous close will open.

Initially, the **User** perspective opens when the QMF application is started. If the **User** perspective is closed you can open it, or any perspective that has been closed, in the following way:

Procedure

1. Click the **Open Perspective** toolbar button. You can also select **Window > Open Perspective**. A menu that lists the perspectives that you have opened in the current QMF session is displayed. Select the perspective that you want to open.
2. To see a complete list of available perspectives, select **Other** from the menu. The **Open Perspective** window opens. Select the perspective that you want to open. Click **OK**.
3. The perspective that you have selected opens. A tab is added to the title bar of the window. The tab displays the name of the perspective and its associated button. Clicking the button for the perspective allows you to quickly switch to a perspective from other perspectives in the same window.
4. By default, a perspective will open in the same window. You can open a perspective in a new window by selecting **Window > Open New Window**. The active perspective opens in a new window.

Closing perspectives

You can close each perspective that is active in the workstation window.

About this task

If you close a perspective, it will stay closed until you re-open that perspective. Each time a QMF application interface is started, all perspectives that were open at the time of the previous close will open, all perspectives that were closed will stay closed.

To close a perspective:

Procedure

1. Select **Window > Close Perspective**. The active perspective closes.
2. To close all perspectives, select **Window > Close All Perspectives**. All of the perspectives that are open in the workstation window are closed. The workstation window appears empty.

Opening views

You can change the layout of a perspective by opening and closing the views that are displayed in the workstation window.

About this task

To open a view:

Procedure

1. Select **Window > Show View**.
2. From the list of views, select the view that you would like to open. Choose **Other** to see additional views. The **Show View** window opens. Select the view that you want to open.
3. Click **OK**. The selected view opens in the perspective that is active in the workstation window. It will be opened automatically with the perspective in subsequent QMF sessions.

What to do next

Note: Views also have their own menus. To open the menu for a view, click the icon at the left end of the view's title bar. Some views also have their own toolbars. The actions represented by buttons on view toolbars only affect the items within that view.

Positioning views

If you are using QMF for Workstation, you can change the layout of a perspective by opening and closing views and by docking them in different positions in the workstation window.

About this task

When opened, a view might appear by itself, or stacked with other views in a tabbed notebook. You can change the location of a view in one of the following ways:

- Drag the view by its title bar to the new location and release the left mouse button.
- Right-click the title bar of the view that you want to move. Select **Move > View** or **Move > Tab Group** from the pop-up menu.

Configuring perspectives

In addition to configuring the layout of a perspective in the application interface, you can customize the options that are available from the **Window** menu; as well as the buttons and menu options that are available from the toolbar and the menu bar.

About this task

The aspects of a perspective that you can control:

- The buttons available on the toolbar.
- The options available in the menus of the menu bar.
- The command groups of the perspective.
- The shortcuts that are available in the perspective.

To configure a perspective:

Procedure

1. Switch to the perspective that you want to configure.
2. Select **Window > Customize Perspective**.
The **Customize Perspective** window opens.
3. Select the tab that contains the options that you want to customize.
4. Expand the item that you want to customize.
5. Use the check boxes to select the elements that you want to see on drop-down menus in the selected perspectives. Items that you do not select will still be accessible by clicking the **Other** menu option.
6. Click **OK**.

Resetting perspectives

If you are using QMF for Workstation, you can restore a perspective to its original layout.

About this task

To restore a perspective to its original layout:

Procedure

1. Select **Window > Reset Perspective**. An informational window opens asking whether you want to reset the current perspective to its default values.
2. Click **OK**. The current perspective is reset to its default values.

Setting preferences

The **Preferences** dialog is used to set user preferences that will apply to specific QMF objects and functions.

The **Preferences** window consists of two panes. The left pane displays the list of preferences groups and the right pane displays the page for the selected group. The following groups of preferences are displayed in the **Preferences** window.

Setting general preferences

You use the **General** page of the **Preferences** window to specify the language, orientation direction, locale settings, and object default actions.

About this task

To specify the general user preferences for the current QMF application interface session:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **General** from the tree.
2. **Conditional:** If available as an option on the **Preferences** window, select from the **Language** drop-down list of supported languages the language that will be used for the current application interface session. Select the value **Auto** to use the local language of the current machine.
3. Select from the **Direction** drop-down list the orientation that will be used to display information in the current QMF application interface session. You can select **Left-to-Right** or **Right-to-Left**. You can select **Auto** to use the default orientation of the current machine.
4. Select from the **Locale setting for numbers, currencies, times, and dates** list the language that will be used to display numbers, currencies, times and dates in the current QMF application interface session. Select the value **Auto** to use the local language of the current machine.

5. Click one of the **Default Action for Objects** radio buttons to specify the action that QMF will perform when you double-click an object that is included in an explorer tree. Select **Run object** to run the object that is selected. Select **Display object** to display the SQL content of the object.
6. In the **Default Tab for the Query Editor** area, you can specify the tab on which the Query Editor is opened by default.
7. Click the **Line width** ellipsis (...) or enter the value of the maximum line width.
The available values for line width are from 10 to 1000. The default value is 79.
This value is used in the **Query** menu item **Reformat Text** for SQL text.
8. Click **Apply** to save your preferences choices.
9. Click **Restore Defaults** to restore the default preference values.
10. Click **OK** to close the **Preferences** window.

Setting appearance preferences

You use the **Appearance** page of the **Preferences** window to specify default fonts, color scheme, and strings for displaying and entering special values. On the page, you can also select saving custom window size.

About this task

To set the appearance preferences:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **Appearance** from the tree.
2. In the **Default Fonts** area, define the font type for headings and columns of query results, classic reports.
3. Specify a string in the **Display null values as** field that QMF will use to display null values that are retrieved from a database data source.
4. Specify a string in the **Enter null values as** field that you will use when you enter null values in the query results or in database tables using the table editor.
5. Specify a string in the **Enter default values as** field that you will use when you enter default values in the query results or in database tables using the table editor.
6. Select from the **Color scheme in text editors** drop-down list what type of color scheme will be supported in the text editors. In most instances you will choose **QMF** to specify that the default color schemes used by the text editors are acceptable. If you will be working in high contrast mode, you will need to select **system default**.
7. If you use a high definition display, select the **Display large icons** check box in the **Icon Size** area to double the size of all icons.
8. If you customize the size and position of the windows in the application and want to save these changes, in the **Window Size** area, select **Save custom window size and position** check box. To revert your changes, clear the check box.
9. Click **Apply** to save your preferences choices.
10. Click **Restore Defaults** to restore the default preference values.
11. Click **OK** to close the **Preferences** dialog box.

Setting dashboard runtime preferences

You use the **Dashboard Runtime** page of the **Preferences** dialog to specify the Google Map API key.

About this task

To specify the dashboard runtime user preferences for the current QMF application interface session:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **Dashboard Runtime** from the tree.
2. Set values for the **Dashboard Runtime**:
You can obtain the Google Map APIs from the Google Code web site.
 - In the **Google Maps JavaScript API key** field, specify the Google Maps JavaScript API key to enable the usage of the Google Map layout object in visual dashboards in the HTML5 rendering mode.
You can obtain the Google Maps JavaScript APIs from the Google Developers web site.
 - Set the **Display null values as** to specify how null values are displayed in the dashboards.
3. Click **Apply** to save your preferences choices.
4. Click **Restore Defaults** to restore the default preference values.
5. Click **OK** to close the **Preferences** window.

Working with global variables

Global variables are variables that stay active while the current session of QMF is active. This is in contrast to substitution variables that are active only during the execution of an object (query, form, procedure).

For objects that use substitution variables, values are entered via a window when the object is run. After the object runs, the variable and the value that you used no longer exists. For objects that use global variables, the value currently defined for the global variable is used.

There are two types of global variables:

- *System global variables* are pre-loaded with the QMF application. That means their values are re-initialized each time you open the QMF application. System global variables begin with the letters DSQQW, DSQAO, DSQEC, DSQDC, and DSQCP. You cannot add or delete system global variables. However, you can edit the default values for the system global variables.
- *User global variables* are defined by the user. User global variables can be specified with any unique name that does not begin with the letters DSQQW, DSQAO, DSQEC, DSQDC, and DSQCP. User global variables values are not saved when you terminate the QMF session.

Using the **Global Variables** page of the **Preferences** window you can view, add, delete and edit your user global variables. You can also edit the values of the system global variables.

Specifying user global variables

You use the **Global Variables** page of the **Preferences** window to view, add, delete and edit your user global variables.

About this task

To view, add, edit and delete user defined global variables:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **Global Variables** from the tree. The **Global Variables** page opens.
2. The system and user global variables that are defined for your session are listed in the **Variables** list box. The name of the global variable is listed in the **Name** field. The current value for the global variable is listed in the **Value** field.
3. To add a new user defined global variable, click the **Add** button. The **New variable** window opens. Type the name of the new user global variable in the **Variable name** field. You can type any combination of unique characters that have not already been used. You can not begin a user global variable with the system global variable prefixes of **DSQQW**, **DSQAO**, **DSQEC**, **DSQDC**, and **DSQCP**. Click **OK**. The user global variable is added to the list and will be available for use during the current

QMF session. To specify the initial value of the variable, edit the **Value** cell as described in the following step.

4. To edit the value of a user global variable, click in the **Value** field for the global variable that you want to change. Type the new value for the global variable in the **Value** field.
5. To delete a user global variable, select an existing user global variable from the **Variables** list and click the **Remove** button. The user global variable is deleted.
6. Click **Apply** to save your preferences choices.
7. Click **Restore Defaults** to restore the default preference values.
8. Click **OK** to close the **Preferences** dialog.

Editing system global variable values

You use the **Global Variables** page of the **Preferences** dialog to edit any system global variable values.

About this task

To edit system global variable values:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **Global Variables** from the tree. The **Global Variables** page opens.
2. The system global variables are listed in the **Variables** list by name. System global variable start with the prefix of **DSQQW**, **DSQAO**, **DSQEC**, **DSQDC**, or **DSQCP**.

Restriction: System global variables whose names start with **DSQAO** cannot be edited.

3. The current system global variable values are listed in the **Value** field. To edit the value of a global variable, click in the **Value** field for the system global variable that you want to change. Type the new value for the system global variable in the **Value** field.
4. Click **Restore Defaults** to restore the default values for all the system global variables.
5. Click **Apply** to save your preferences choices.
6. Click **OK** to close the **Preferences** window.

Setting help preferences

If you are using QMF for Workstation, you use the **Help** page of the **Preferences** window to specify the default values for how help information is displayed and to change the interface and port specification for the internal server that the help system uses.

About this task

To change the web browser adapter setting:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **Help** from the tree.
2. The help system can display information in either the **Help** view or in a browser. If the help is displayed in a browser, the browser can be an embedded browser or a full external browser. If the embedded browser is supported on your system, help will use it by default to display help. If you prefer to always use a full external browser, select **Use external browser** from the drop-down list.
3. By default, context-sensitive help for workstation windows is displayed in the **Help** view. If you prefer to display workstation window context-sensitive help in infopops, select **Open window context help in an infopop** from the drop-down list.
4. By default, context-sensitive help for dialogs is displayed in the dialog's tray, which is similar to the **Help** view. If you prefer to display dialog context-sensitive help in infopops, select **Open dialog context help in an infopop** from the drop-down list.

5. Click **Apply** to save your preferences choices.
6. Click **Restore Defaults** to restore the default preference values.
7. Click **OK** to close the **Preferences** window.

Including help content from a remote server

The help system includes an internal server that serves the help contents to the browser. You use this preference page to change the interface and port that the internal server uses.

About this task

You should only change these settings if you experience problems and cannot view help with the default preferences.

To change the interface and port that the help server uses:

Procedure

1. Select **View > Preferences**. The **Preferences** dialog opens. Expand the **Help** branch by clicking the plus sign. Select **Content** from the tree.
2. Select the **Include help content from a remote infocenter** check box.
The **Remote Infocenter** window becomes available.
3. Click **Add** .
The **Add new infocenter** window opens.
4. Specify the name of a local IP interface to be used by the server in the **Name** field.
5. Specify the host of a local IP interface to be used by the server in the **Host** field.
6. Specify the URL of a local IP interface to be used by the server in the **Path** field.
7. Select **Use port** and specify the number of a specific port the server will listen to.
8. Click **OK**.
The **Add new infocenter** window closes and control returns to the **Preferences** window. The new infocenter is displayed in the **Remote Infocenters** window.
9. To edit an infocenter, click **Edit** .
The **Edit [infocenter name] infocenter** window opens.
10. To delete an infocenter, click **Delete** .
The selected infocenter is removed from the list.
11. To view the properties of a specific infocenter, click **View Properties** .
The **Properties for [infocenter name]** window opens with the name, host, path, port, and URL displayed.
12. To test an infocenter's connection to the local IP interface, click **Test Connection** .
The **Test Connection** window opens displaying the test results.
13. To disable an infocenter without deleting it, click **Disable** .
The infocenter is now disabled, and its status in the **Enabled** column is changed to disabled.
14. Click **Apply** to save your preferences choices.
15. Click **Restore Defaults** to restore the default preference values.
16. Click **OK** to close the **Preferences** window.

Setting JDBC preferences

You can use the **Preferences** window to set up the JDBC driver information.

About this task

QMF administrators are responsible for distributing the JDBC driver information that users will need in order to access any data source. In most cases, administrators distribute this information by deploying a preconfigured JDBC library settings file when the application interface that you are using is installed.

Advanced users can define or change JDBC driver location information using the **JDBC Libraries** page of the **Preferences** window to specify the JDBC driver information for each type of database (such as Db2®, Informix, Oracle) that will be accessed. The JDBC driver locations must be accessible to QMF. QMF will retain the JDBC driver information. You would only repeat this specification process to add new drivers or make changes to the properties of an existing driver.

To define JDBC driver information:

Procedure

1. Select **View > Preferences** to open the **Preferences** window. Select **JDBC Libraries**. The **JDBC Libraries** page opens. All of the JDBC driver libraries that you have already set up are listed in the **JDBC driver libraries** area. The JAR files associated with selected driver library are listed in the **Driver files** area. Additional information about the selected driver library is displayed on the **Details** panel.
2. To add new JDBC driver, select **Add Driver**. The **Add New Driver Library** window opens.
3. The JDBC driver information for each type of database is organized by libraries. Enter a name for the library in the **Enter new library name** field. It is recommended that the name that you specify describes the JDBC information that will be included in the library. For example, if you are adding JDBC driver information for accessing Db2 databases, you might call the library Db2. However, this is a descriptive field and can include any text.
4. Specify the actual name of the driver class that will be used in the **Driver class name** field. This can be found in your JDBC driver documentation. For example: com.ibm.db2.jcc.DB2Driver.
5. Optional: Specify a generic example of a correctly formatted URL that could be used to connect to the database in the **URL templates** field. For example, if you are adding JDBC driver information for accessing Db2 databases, you might specify the following JDBC URL template: jdbc:db2://host:50000/database. The generic information as specified in the template is presented when you are adding data sources to the repository and when you are creating repository connections. You will replace the generic information with the specific database information.
6. Click **OK** to create the driver library and close the **Add New Driver Library** window.
7. If you are using QMF for Workstation, add the location information for the JDBC driver files to the library. If you are using QMF for Workstation, click **Add Driver Files**. The **Add Files to [libraryname]** window opens. Search for and select the JDBC driver files that you want to add. Click **Open**. The location of the JDBC driver files is saved in the JDBC library that you have selected.
8. You can make the following changes to the JDBC driver information:
 - To edit the JDBC driver library information, select an existing JDBC driver library from the list and click **Edit Driver**. The **Edit Driver Library** window opens where you can make changes to the **Library name**, **Driver class name**, and **JDBC URL template** fields.
 - If you are using QMF for Workstation, to edit the name and location of the files that contain the JDBC drivers, select a file from the list of files that have been added to the JDBC library, and click **Edit**.
 - To remove a JDBC driver library, select an existing JDBC driver library from the list and click **Remove Driver**. The library, including all the JAR files that it contains, is removed.
 - If you are using QMF for Workstation, to remove a JAR file from a JDBC driver library, select an existing file from the list and click **Remove**. To remove a JAR file from a JDBC driver library, select an existing file from the list and click **Remove Driver File**. The file is removed.

9. Click **OK**. The JDBC library information is saved. The **Preferences** window closes. You must repeat this process for each type of JDBC driver that will be used to access various database data sources such as Informix or SQL Server.

Related tasks

[Importing the JDBC library settings](#)

You can use the JDBC library settings file to set up the JDBC driver information.

Setting LOB preferences

You use the **LOBs** page of the **Preferences** dialog to specify options that will control retrieving and saving LOB data.

About this task

LOB data (Large Object data) is large text or binary objects in a database. LOBs can be large strings of text, images, and so on. If a query result set contains a large amount of LOB data, processing resources could be severely strained. The **LOB** preferences page contains a number of LOB data retrieval restrictions that you can use to manage repository resources. To specify LOB data preferences:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **LOBs** from the tree.
2. Check **Override LOB options if possible** to grant yourself the ability to override the LOB options that have been specified by your QMF administrator for your resource limits group. Your administrator must have given your resource limits group permission to override the LOB options. If you do not have permission to override LOB options, this check box is not available.
3. Use the **LOB Retrieval Option** radio buttons to specify how large object data (LOB) will be retrieved. There are four choices:

- Select **Disable LOB columns** to disable LOB data retrieval. By selecting this option, you will not be able to query any table that contains LOB data.
- Select **Disable LOB data retrieval** to disable LOB data retrieval but allow access to the other columns in the tables that contain LOB data. By selecting this option, you can query tables that contain LOB data and result data for all columns except those that contain LOBs will be returned.
- Select **Retrieve LOB data on demand** to specify that you want to retrieve all LOB data from a table but only display selected columns of the LOB data in the query results.

By selecting this option, you can query tables that contain LOB data and your result data will be returned for all columns that contain LOB objects. To conserve resources, however, the retrieved LOB data is stored in a file. Pointers (<LOB LOCATOR>) to the LOB data objects are displayed in the query results. You click the pointer to view the LOB data.

You can also pull the data from the database by double-clicking on the pointer.

- Select **Retrieve LOB data automatically** to specify that you want to retrieve all LOB data from a table and immediately display the retrieved LOB data in the query results.

This option pulls ALL of the LOB data for ALL of the LOB columns from the database to the local computer. The actual LOB data is not displayed in grids and reports. Instead, pointers to the LOB data are displayed.

This option can potentially consume a large amount of resources on the local computer.

4. Use the **LOB Saving Option** radio buttons to specify whether you can save large object data (LOB). There are two choices:
 - Select **Disable LOB data** to disable saving LOB data on a database data source.
 - Select **Enable LOB data** to allow saving LOB data on a database data source.
5. Specify the maximum size of a LOB column in the **Maximum LOB column size** field. You specify the size in kilobytes, up to 2G (the maximum LOB size). The default is 0, which specifies there is no

maximum size. If you query a table with LOB data that is larger than the maximum, the LOB data will not be returned for display.

6. Click **Apply** to save your preferences choices.
7. Click **Restore Defaults** to restore the default preference values.
8. Click **OK** to close the **Preferences** window.

Setting log preferences

You use the **Log** page of the **Preferences** window to activate a log file that will track QMF processing information.

About this task

The log file information can be useful in debugging. It can be delivered as a report or to your workstation in the **Output** view.

To specify the log file preferences for the QMF session:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **Log** from the tree.
2. Check **Enable log** to activate the log file for debugging purposes and deliver the information in the **Output** view of your workstation. If this check box is selected, the log file option fields are enabled. Check one or more of the log file options to indicate what information should be gathered. It is recommended that all options remain checked. The available log file options are as follows:
 - Print stack trace for log exceptions
 - Print report to log file
 - Print log class and method
3. To specify that additional information be gathered, click **Edit Log Categories**. The **Edit Logger Categories** window opens. Select the additional information that should be collected. Click **OK**.
4. Check **Enable report** to deliver the processing information in report format. If this check box is selected, the report file option fields are enabled. Check one or more of the report file options to indicate what information should be gathered. The following check boxes are the report file options:
 - Print stack trace for report exceptions
 - Print report time
 - Print report class and method
 - Automatically activate **Output** view
5. To specify that additional information be gathered, click **Edit Report Categories** button. The **Edit Report Categories** window opens. Select the additional information that should be collected. Click **OK**.
6. Click **Apply** to save your preferences choices.
7. Click **Restore Defaults** to restore the default preference values.
8. Click **OK** to close the **Preferences** window.

Setting the search configuration

You use the **Search Configuration** page of the **Preferences** window to modify your search preferences.

About this task

You can enable or disable search on QMF objects, View, Indexing location, specify your search output limit, specify the target categories on which search is performed.

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **Search Configuration** from the tree.
2. From the **Indexing** options, you can select **Index On** or **Index Off**.
Indexing is the basic requirement of the search functionality. Only indexed objects are searched. When you select index off, the newly created objects are not indexed and hence are not searched. Since search operates only on the indexed objects, we recommend to select the **Index On** option for the search to work smoothly and return correct and updated results. For example, if you move a file from one folder to another, the index functionality updates the new file location and allows search to be performed on the file in the new location. The new changes to the objects such as creating a new object, updating it, or deleting it, are automatically updates the index.
3. **Storage Location**: Storage location is the location where the indexes are created. Storage location is not editable.
4. **Results Limit**: This defines the number of objects returned in the search result. By default, result limit is set to five, which means only five objects are displayed on the screen.
5. **Object List**: This is the target category on which the search is performed. You can enable connected **Repository**, **Workspaces**, and **Query Content** categories for your search. By default, the search on **Query Content** is not enabled.

Setting SMTP server preferences

You use the **SMTP** area of the **Preferences** window to specify the list of available SMTP servers.

About this task

Defining parameters for commonly used SMTP servers in the SMTP area saves users from having to specify these parameters every time they need to send an email message by using these servers. For example, if the SMTP server password is expired, you just need to change it once in the SMTP area instead of updating authorisation information for every object that uses this SMTP server.

The servers listed in the SMTP area can be used in the following procedures:

- Adding a Send Mail action to a Visual Project
- Sending email messages by using the **File > Send to** command
- Sending email messages by using the MAIL TO procedure command.

To specify SMTP server preferences:

Procedure

1. Select **View > Preferences** to open the **Preferences** window. Select **Email** from the tree and navigate to the **SMTP** area.
2. Add a new server to the list by clicking **Add Server**.
3. In the **SMTP** window specify SMTP server parameters including its name, address and port number.
4. Optional: If the SMTP server requires authentication, select **Use SMTP authentication**, and then specify user name and password in the **User** and **Password** fields.
5. Optional: If it is necessary to specify additional access preferences, click **Advanced** to open the **Advanced SMTP Settings** window where you can add required keywords supported by `javamail`. Click **OK**.
6. Click **OK** to close the **SMTP** window and then click **Apply** to save your preferences choices.
Tip: Select or clear the **Enabled** check box to control the SMTP settings globally.
7. Optional: If you need to restore default preferences, click **Restore Defaults**.
8. Click **OK** to close the **Preferences** window.

Setting the directory location preference for objects generated by Export actions

You can set preferences in QMF so that objects generated from various QMF actions are exported to a specific location on either the local file system or the server file system.

Before you begin

If you are using QMF for Workstation, create a folder on the local file system.

If you are using QMF for WebSphere, create a folder on application server file system.

When you set the **Server-Side File System** preference, you will point to one of these folders.

About this task

This task describes how to specify (through preference settings) the location to which generated objects will be exported.

To set the export directory preference:

Procedure

1. Select **View > Preferences** and select **Server-Side File System** from the tree.
2. Select the **Enable access to the server file system** check box.
3. Type the path in the **Root output directory** field to indicate where generated objects will be exported.

When data and objects are generated by the **EXPORT** procedure, scheduled tasks (**Copy** or **Execute** object) or by event actions such as **Export to PDF** and **Export to Excel**, they will be exported to path you specify.

The directory you specify must exist on either the local file system or on the server file system.
4. Type the path in the **Root input directory** field to indicate the path of the root directory for dynamart source files.

Note: You need to specify a **Root input directory** path, otherwise the settings are incorrect, and the **EXPORT** using server-side file system will yield an error.
5. Click **Apply** to save your preferences.
6. To restore preferences to the default values, click **Restore Defaults**.
7. Click **OK** to close the **Preferences** window.

Setting root directory location for dynamart source files

You can specify a root directory location where dynamart source data files are stored.

About this task

When you save a data file as a dynamart, the link to this file is saved within the dynamart. When you open this dynamart, QMF reloads the data from the linked source file and updates the result set if the source file was modified or replaced.

You can specify the Server-Side File System preferences so that the data files that are linked to dynamarts are stored in a specific location on either the local file system or server file system.

Note: You can change the source data file or disable updating of data in the **Dynamart Setting** window.

Procedure

1. Select **View > Preferences** and select **Server-side File System** from the tree of preferences.
2. Select the **Enable access to the server file system** check box.

3. Type the path of the root directory for dynamart source files in the **Root input directory** field.

Note: The directory you specify must exist on either the local file system or on the server file system.

4. Click **Apply** to save your preferences.
5. To restore preferences to the default values, click **Restore Defaults**.
6. Click **OK** to close the **Preferences** window.

Setting third party library preferences

You use the **Third-Party Libraries** page of the **Preferences** window to specify the location of any third-party software that is required by QMF.

About this task

To specify the third party library preferences:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **Third-Party Libraries** from the tree.
2. The third-party software whose location has been defined to QMF is listed by name, package name and location. In most instances, the location information for third party libraries is specified when QMF is installed.
3. To add additional location information for third party software, click **Add**. The **New Third-Party Library** window opens. Specify the path to the software in the **Path** field. If applicable, specify the packages name in the **Packages (optional)** field. Click **OK**. The path to the software is added to the **Third party library** list. Control returns to the **Preferences** window.
4. To edit the location information for defined third party software, select **Edit**. The **Edit Library** dialog opens. You can change the path to the selected software in the **Path** field. You can change the packages name in the **Packages (optional)** field. Click **OK**. The changes that you have made are applied. Control returns to the **Preferences** window.
5. To remove third party library information, click the third party library in the **Third party libraries** list and click **Remove**. The third party library information is removed from the list.
6. Click **Apply** to save your preferences choices.
7. Click **Restore Defaults** to restore the default preference values.
8. Click **OK** to close the **Preferences** window.

Specifying Visual Designer preferences

If you are using QMF for Workstation, you use the **Visual Designer** page of the **Preferences** window to specify preferences that will effect the creation of visual projects.

About this task

To specify the preferences that will effect the creation of visual projects:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **Visual Designer** from the tree.
2. Click one of the **Visual Designer Ruler Unit** radio buttons to specify the default units that will be used for the ruler that spans the editor window in the **Visual Designer** perspective. Your choices are **pixels**, **inches**, or **centimeters**.
3. Select a formatting schema for date values of a classic report in the **Formatting for &DATE values** drop-down list.
4. Select a formatting schema for time values of a classic report in the **Formatting for &TIME values** drop-down list.

5. Select the **Reset global variables when entering runtime mode** check box to reset the global variable values back to default values each time a visual project is run.
6. Expand **Visual Designer** in the tree and select **Fonts**.
7. To make a different set of fonts available for visual dashboards, specify the path to a custom set of fonts in the **Search directory** field and click **Refresh Associations**.
8. Select **Project Explorer** from the tree.
9. Specify a specific number of times that you can undo an action in the design editor in the **Undo limit** field.
A value of '0' indicates that there is no limit to the number of times that you can undo an action.
10. In the **Rename preferences** radio group, specify whether you will be prompted to confirm when you attempt to rename an object in the design editor.
The available options include:
 - **Always prompt** - You will always be prompted when you attempt to rename an object.
 - **Prompt when referenced** - You will only be prompted when the object has references to other objects.
 - **Never prompt** - You will never be prompted when you attempt to rename an object.
11. In the **Delete preferences** radio group, specify whether you will be prompted to confirm when you attempt to delete an object in the design editor.
The available options include:
 - **Always prompt** - You will always be prompted when you attempt to delete an object.
 - **Prompt when referenced** - You will only be prompted when the object has references to other objects.
 - **Never prompt** - You will never be prompted when you attempt to delete an object.
12. Click **Apply** to save your preferences choices.
13. Click **Restore Defaults** to restore the default preference values.
14. Click **OK** to close the **Preferences** window.

Exploring the Workspaces view

All of the data sources and objects that you can access are contained in one or more workspaces that have been pre-populated for you by the QMF administrator.

About this task

Each workspace to which you have access is listed in the **Workspaces** view. From the **Workspaces** view you can perform the majority of query and reporting functions.

Discovering the contents of a workspace

The **Workspaces** view displays the contents of a workspace.

About this task

To discover the contents of a workspace:

Procedure

1. To open a workspace, click the workspace name in the **Workspaces** view. The content of the workspace is displayed in a tree structure.
2. You can expand any node of the tree by clicking the plus sign (+) to the left of the node name. You continue to expand folders and directories to uncover the contents of the workspace. Many object types are listed in your workspace. The type of object is identified with a unique icon.

3. If you right-click anywhere in the **Workspaces** view, or on an object that is listed in the node, a pop-up menu opens where you can select from one of the available functions.
4. Each data source that has been configured in your workspace is listed and identified with the data source icon. To expand a data source, click the plus sign (+) to the left of the data source name. The **Login** window opens where you must specify the user ID and password that will be used to login to the data source.
5. Each data source that is listed has one or more of the following folders depending on how the data source was configured:
 - The **Database** folder which includes the native objects (procedures, schemas and tables) that are on the data source.
 - The **QMF Catalog** folder which lists the objects (forms, queries, procedures, and schemas) that are included in the QMF catalog. This folder is only included if the data source has been configured to access a QMF catalog.
 - The **OLAP** folder lists the available OLAP cube models. This folder is only included if the data source has been configured to enable OLAP querying.
6. To open an object that is listed in the **Workspaces** view, double-click that object. Depending on what you have set as your default action, double-clicking on any object listed in your workspace will either run or open the object. The default action for objects is set using **Default Action for Objects** found on the **General** page of the **Preferences** window. **Default Action for Objects** control the action for all objects other than visual reports and dashboards.
7. If your user ID has permission, you can add folders to your workspace and move objects into the folders, organizing your information according to your requirements. The QMF administrator specifies which users have permission to add folders, move objects and otherwise edit the contents of the workspace.

Workspace menu commands

From the **Workspaces** view, you can right-click a node, folder or object and open a pop-up menu. The menu choices depend on what you have selected.

New

Opens a pop-up menu where you can launch one of the many wizards that are available to help you create objects and other repository items.

Open

Opens the object that you have selected in the **Workspaces** view. The table is run and the results are returned in the editor window.

Open From

Opens a menu where you can choose to open an object that has been saved in a:

- Repository
- File

Open With

Opens any database table that is accessible to you in your workspace using a default editor, the Table Editor, the Table Viewer or the Visual Query Editor.

Run

Runs the object that you have selected in the **Workspaces** view. The results of running the object are returned in the editor window.

Run As

Runs the visual project that you have selected in the **Workspaces** or **Repository Explorer** views in a specified rendering mode.

Set Name Filter

Opens the **Filter by Name** window where you can specify the names of the objects that you want to list for the selected data source in the **Workspaces** view. Using a percent sign (%) will include all objects.

Set User Information

Opens the **Filter by Name** window where you can specify the names of the objects that you want to list for the selected data source in the **Workspaces** view. Using a percent sign (%) will include all objects.

Rename

Renames the object or folder that you have selected in the **Workspaces** view.

Delete

Deletes the object or folder that you have selected in the **Workspaces** view.

Copy

Copies the object or folder that you have selected in the **Workspaces** view and writes it to the clipboard.

Paste

Pastes an object or folder from the clipboard to the parent folder or node that you have selected in the **Workspaces** view.

Paste Link

Pastes the path to the object or folder that is currently on the clipboard into the folder or node that you have selected in the **Workspaces** view.

Paste as Offline

Pastes a true copy of the object or folder that is currently on the clipboard into the folder or node that you have selected in the **Workspaces** view.

Add to Favorites

Adds the object that you have selected to the **Favorites** node in the **Personal** view.

Add to Startup

Adds the object that you have selected to the **Startup** node in the **Personal** view.

Explore

Opens a window in the **Editor** view that is formatted in an explorer format. Objects, folders, and nodes can be filtered in this window.

Refresh

Refreshes the contents of the **Workspaces** view that is currently displayed. The **Workspaces** view is completely refreshed regardless of what node is selected.

Properties

Opens the **Properties** window for the currently selected object, folder, data source, or node. The **Properties** window provides specific information about your selection.

Show in Repository

Displays a dialog that shows all objects that reference the selected object in the **Repository Explorer** view.

Working with QMF search

You can search on QMF objects using the search functionality.

About this task

You can open the search by clicking on the search icon on the toolbar or by clicking on the **QMF Search** from **Window > Show View > Other > QMF Workbench**.

The **QMF Search** window pane opens and you can type the text to be searched in the **Search** field. This operation returns search results.

When you click on a search result, the QMF object is highlighted in the expanded **Repository Explorer** tree and opens in the editor.

The search functionality allows you to search the QMF objects with a specific file name or specific file content. This search operation is performed across multiple data sources within the connected repository. You can also search for the content of Visual Query and Procedures.

QMF objects are indexed while user performs operations such as create, edit, move, or delete objects. Search functionality searches only indexed objects.

QMF supports the following types of search:

- **Wildcard search:** If you want to search specific characters, you can concatenate * to those characters. Search will return all the QMF objects starting or ending with those characters.
- **Fuzzy search:** If you want to find strings that approximately match a pattern in your data, you can perform fuzzy search by appending ~ in the search text.
- **Proximity search:** Proximity searches allows you to find terms that are near each other in a document. Insert a tilde “~” symbol at the end of a phrase followed by the number of words that create the proximity boundary. For example, "select mytable"~5 will find the terms “select” and “mytable” within 5 words of each other.

Search is supported on the following objects:

- Ad hoc Report
- Analytical Query
- Dynamart
- Folder
- Forecast
- Form
- Procedure
- Quick Report
- Relational Data Source
- Visual Dashboard
- Visual Query
- Visual Report
- Workspace

Managing workspace folders

If your user ID has permission, you can add folders to your workspace and move objects into the folders, organizing your information according to your requirements.

About this task

The QMF administrator specifies which users have permission to add folders, move objects and otherwise edit the contents of the workspace.

To manage repository folders:

Procedure

1. From the **Workspaces** view, right-click the node or existing folder that you want to be the parent of the new folder. From the pop-up menu, select **New > Folder**.
The **Create New Folder** window opens.
2. Type the name of the new folder in the **Name** field.
3. Specify the parent folder of the new folder in the **Parent folder** field.
4. Optionally, you can type a comment that will describe the folder in the **Comment** field.
You can view the added comment when you view the **Properties** of the folder.
5. Click **Finish** to add the folder.
6. To delete a folder, right-click the folder in the **Workspaces** view. From the pop-up menu, select **Delete**.
The folder and all of its contents are deleted.

7. To rename a folder, right-click the folder in the **Workspaces** view. From the pop-up menu, select **Rename**. The folder is renamed.

Filtering data source objects

If you are using QMF for Workstation, you can filter by name the objects that will be displayed for a data source in the **Workspaces** view.

About this task

To filter data source objects:

Procedure

1. From the **Workspaces** view, right-click a data source that you want to filter. From the pop-up menu, select **Set name filter**. The **Set Name Filter** window opens.
2. In the **Object owner** field, specify the identifier that owns the objects that you want to include in the **Workspaces** view when the data source is opened. You must specify the owner in the correct format for identifiers. You can use a pattern with wild cards in this field to match multiple owners.
3. In the **Object name** field, specify the identifier that names the objects that you want to include in the **Workspaces** view when the data source is opened. You must specify the name in the correct format for identifiers. You can specify a matching pattern to match multiple names.
4. Click **OK** to close the **Set Name Filter** window. Control returns to the **Workspaces** view. Only the objects that meet the criteria that you specified in the **Object owner** and **Object name** fields are included.

Creating links to workspace objects

You can set up a link to any object that is listed in the **Workspaces** view.

About this task

Links are useful if you want to move several objects into a folder without actually copying the object.

To create a link to a workspace object:

Procedure

1. Right-click the object in the **Workspaces** view. Select **Copy** from the pop-up menu. The object and the path to the selected object are saved on the clipboard.
2. Right-click the folder that you want to be the parent folder for the link information to the object. Select **Paste link** from the pop-up menu. The link to the object is listed in the **Workspaces** view.
3. To delete an object link, right-click the link in the **Workspaces** view. From the pop-up menu, select **Delete**. The link is deleted.
4. To rename a link, right-click the link in the **Workspaces** view. From the pop-up menu, select **Rename**. The object link is renamed.

Creating links to repository objects

You can set up a link to any object that is in a repository to which you have access.

About this task

Links are useful if you want to access an object without copying the object to your workspace.

To create a link to a repository object:

Procedure

1. Right-click anywhere in the **Workspaces** view to open the pop-up menu. Select **New > Other > Repository > Link**. The **Create New Link to Repository Object** wizard opens.
2. Specify the name for the link information in the **Name** field. This is the name that will be listed in the **Workspaces** view.
3. Specify the path including folder name to the parent folder in the **Parent folder** field. You can use the **Browse** button to search for the folder.
4. Specify the object for which you want to create a link in the **Linked object** field. You can use the **Browse** button to search for the object.
5. Type any text that will be used as comment information in the **Comments** field. Comment information is displayed in the **Properties** view for the link.
6. Click **Finish**. The **Create New Link to Repository Object** wizard completes its processing.

Viewing properties for workspace objects

The **Properties** dialog displays the property values for a selected object from the **Workspaces** view.

About this task

The information provided varies based on the object that is selected. Objects, folders, data sources, and workspaces all have property values that can be displayed. From the **Properties** dialog you can also modify some property values.

To view or change property values:

Procedure

1. Right-click an item in the repository, select **Properties** from the pop-up menu. The **Properties** window opens.
2. The content of the **Properties** window varies based on the type of item that you select. For all items in the workspace an information page is available. This page gives detailed information on the workspace object. You can add or edit comment information on this page.
3. Additional property pages are available for data sources that are included in the workspace. Select one of the following from the tree:
 - **Connection Parameters:** To view or edit the connection information that is used to connect to the data source.
 - **Plug-ins:** To view or change the plug-ins that have been activated for the data source.
 - **Resource Limits:** To view, define, or edit the resource limits that have been defined for the data source.
 - **Login mapping:** To view, define, or edit the login mapping IDs that have been defined for the data source. (QMF for Workstation only)
 - **Security:** To view, define, or change the users and groups that can access the data source, if it is under the control of the **Internal** or **LDAP** security option. You can also view or change the permission levels that have been granted to each user or group. (QMF for Workstation only)
4. To restore all the default values for the property values, click **Restore Defaults**.
5. To implement the changes that you have made to any of the property values, click **Apply**.
6. Click **OK** to close the **Properties** window.

Refreshing workspace contents

You can refresh the contents of the workspace. Refreshing updates any repository objects with the latest versions.

About this task

Every time you open an object, the workspace automatically refreshes. You must refresh the workspace if another user saved an object to the same workspace from another session of the application. You can refresh the contents of the **Workspaces** view in one of the following ways:

- Right-click anywhere in the **Workspaces** view. Select **Refresh** from the menu.
- From the **Workspaces** view toolbar, select the **Refresh Workspaces** button.

Moving workspace objects

If you are using QMF for Workstation, using your mouse, you can drag and drop objects and folders that are listed in the **Workspaces** view.

About this task

To move workspace objects:

Procedure

1. Click an object from the **Workspaces** view.
2. With the mouse button pressed, drag the item to its target location. Release the mouse button. A pop-up menu opens.
3. Select one of the following:
 - **Copy Here** to copy the object to the target location. A copy is made. The original remains.
 - **Move Here** to move the object to the target location. The original copy is moved to the new location.
 - **Create Link Here** to create a link to the object.

Adding content to a workspace

After you create a workspace, you then add content to the workspace.

About this task

If you want to access a data source from the workspace you must add the information necessary to access the data source. If you want objects to be available from the workspace you must add the information necessary to access the object. For both data sources and objects you can create this information as links or offline entries.

To add content to a workspace:

Procedure

1. You can discover data sources and objects that can be added to the workspace in one of the following ways:
 - Open the **Repository Explorer** view. The **Repository Explorer** will display the complete contents of a repository.
 - Open any other workspace to which you have access based on your user ID
2. Select the data source or object that you want to add to the new workspace. Your user ID must have permission to access the data source or object in order to add it to your new workspace.
3. You can add the data source or object to the workspace in one of the following ways:

- With the mouse button pressed, drag the data source or object to the new workspace. When you release the mouse button, a pop-up menu opens. Select **Create Link Here** to create a link to the data source or object. Select **Create Offline Entry** to actually copy the data source or object to the workspace.
 - Right-click the data source or the object. Select **Copy** from the pop-up menu. The location of the data source or object is copied to the clipboard. Right-click the workspace to which you want to add the data source or object. From the pop-up menu select **Paste Link** to create a link to the data source or object. Select **Paste as Offline** to actually copy the data source or object to the workspace.
4. Depending on your selection the information required to access a data source or object from the workspace is added as follows:
- If you selected **Create Link Here** or **Paste Link**, a link to the data source or object is created in the workspace.
 - If you selected **Create Offline Entry** or **Paste as Offline** the data source or object is copied in the workspace. Copying data sources or objects can be time-consuming. However, offline entries allow you to work with an object from a data source without actually accessing the data source. This can save time and cut resource usage.

Creating additional workspaces

You can create additional workspaces in the **Workspaces** view if your user ID has been granted full security permission for the repository in which your workspace resides.

The QMF administrator assigns a user permission to add workspaces when the repository is created. After creating a new workspace you can add data source information and objects to the workspace.

Adding a workspace as a user

If your user ID has full security permission you can add additional workspaces to a repository.

About this task

To add a workspace as a user, perform the following procedure:

Procedure

1. Select **File > New > Other**. The **New** window opens. Expand the **Repository** folder. Select **Workspace**. The **Create New User Workspace** wizard opens.
2. Specify the name for the new workspace in the **Name** field.
3. Specify a comment that describes the workspace in the **Comment** field.
4. Click **Finish**. After you have created a workspace, you specify what data source information and objects will be available in the workspace.

Working in the Personal view

The **Personal** view provides you with quick access to your most frequently or recently used objects. In addition, it supplies a list of objects that will automatically open when the application starts.

About this task

The **Personal** view includes the following folders:

- **Recently Used:** This folder contains a list of all the objects recently opened by the user. This list is populated by the application.
- **Favorites:** This folder contains a list of the user's favorite objects. To add an object to the **Favorites** folder, right-click the object in the **Workspaces** or **Repository Explorer** view and select **Add to Favorites**.

- **Startup:** This folder contains a list of objects that will automatically open upon startup of the **User** perspective. To add an object to the **Startup** folder, right-click the object in the **Workspaces** or **Repository Explorer** view and select **Add to Startup**.

Personal view menu commands

From the **Personal** view, you can right-click a node, folder, or object and open a pop-up menu. The menu choices depend on what you have selected.

Remove from Favorites

Removes the selected object or folder from the **Favorites** node.

Change favorite name

Changes the name of the selected object or folder in the **Favorites** node. This does not change the original name of the object or folder, only its representation in the **Personal** view.

Remove from Startup

Removes the selected object, folder, or node from the **Startup** node.

Change Startup Name

Changes the name of the selected object, folder, or node in the **Startup** node. This does not change the original name of the object, folder, or node; only its representation in the **Personal** view.

Chapter 3. Accessing the data

The data you will use to create BI content exists on your enterprise information systems, in databases. So, before you begin to develop BI content (reports, visual dashboards, queries etc.) you must access the data that serves as the "raw material" for what you will create.

Repositories and data sources

QMF uses repositories as centralized storages of data sources and application objects.

Repositories

QMF supports personal and shared repositories. Personal repositories are local to individual users and shared repositories are located on a database and can be accessed by several users.

To work with the application, users of QMF must connect to a repository. They can have multiple QMF repositories, each of which can contain different data.

Users can access a shared repository indirectly by using web service or network connections.

Data Sources

Data sources are entities that store connection information to access databases. In repositories, data sources are classified by the types of the databases that they represent:

Relational

Data is stored in interrelated tables. Each table comprises a number of columns and rows.

Multidimensional

Data is stored in cubic structures. Each dimension of a cube represents a particular type of data. Databases of this type are more complex than relational ones and provide an opportunity to run queries that are more elaborate.

Hive

Data is stored in Apache Hive™ data warehouses. This kind of storage is used in Apache Hadoop™ software framework designed for summarizing, querying, and analysing of large volumes of data with the help of HiveQL, a language that is very similar to SQL.

Virtual

Data is stored in virtual and JavaScript tables that collect information from different sources and present it as a single database. Virtual databases cache data from original databases so that users can work with it without referring to original data sources separately.

JavaScript

Data is provided by online services and stored in JavaScript tables.

Folder

Data is stored on the local file system.

Regardless of the type of the data source that you use, the application processes the data that is retrieved from any data source as a flat table.

Creating personal repositories

QMF users can create their own personal repositories. Personal repositories are local to the individual user and are generally only used by that individual. Administrators can also create personal repositories on behalf of a user.

About this task

Personal repositories serve a single user and are stored in the personal settings directory of the workstation (server) that is running the QMF application interface.

Personal repositories do not support any of the extended features of repositories such as security.

To create a personal repository, perform the following procedure:

Procedure

1. Select **File > New > Other** . Expand the **Repository** folder, select **Personal Repository** and click **Next**. The **Create New Personal Repository** window opens.
2. Specify the name for the repository in the **Name** field.
3. Optionally, you can specify a comment that describes the repository in the **Comment** field.
4. To connect to the repository immediately, select the **Connect immediately** check box.
5. Click **Finish**. The personal repository is created and saved in the workstation's personal settings directory. Each user is responsible for populating their personal repository. You can save objects in your personal repository.

Connecting to a repository

In order to access a repository and run repository objects using QMF you must be connected to a repository.

About this task

A repository is a centralized storage area that was created by your QMF administrator. It is the place where your objects such as queries, forms, and reports can be saved. It is also where QMF will look for the information necessary to connect to any data sources that you need to access. The **Repository Connections** view lists all of the currently available repository connections.

Each time that you open QMF, an attempt is made to connect to the repository that was last used. At any time, you can connect to a different repository by selecting the repository connection information from the **Repository Connections** view. In addition, you can choose to connect to different repositories using different user IDs and passwords.

To connect to a repository:

Procedure

1. Open the **Repository Connections** view if it is not open in the workstation window. Select **Window > Show view > Repository Connections**. The **Repository Connections** view opens.
2. From the **Server** list, select the QMF Server whose repository connections you want to display, or select **Local** to display the repository connections that are available on your local machine.

If the QMF Server that you have selected from the **Server** list has one or more registered QMF Catalogs, they will be available as a separate connection in the **Repository Connections** view. If you establish this connection, you will be able to access QMF Catalog objects from the **Repository Explorer** view.

Note: QMF Server-based repository connections will be available to you only if you have an instance of QMF configured for your QMF client.

3. Right-click the repository connection name for the repository to which you want to connect.
4. Select **Connect**. You are connected to the repository.

What to do next

Administrators are responsible for creating repository connections for each user of QMF. If necessary, each QMF for Workstation user can create repository connections. Creating repository connections is an advanced function. Most users will not create repository connections. However, some advanced users might create repository connections in some circumstances such as:

- If the repository of your choice is not listed in the **Repository Connections** view, you can create a new repository connection using the **Create New Repository Connection** wizard.
- If the connection information has not been set up for your version by the administrator, you must set up the information that your version of QMF will use to connect to a repository. You will recognize that this is the case if there is no connection information listed in the **Repository Connections** view. Your QMF administrator will provide the repository connection information either directly or by providing a repository connections settings file that you can import.

Specifying user information

You can use the **Set User Information** dialog to change the user ID and Password that a query will use when connecting to a data source.

About this task

The **Set User Information** command is available from the **Query, Procedure, Form** or **OLAP query** menu.

To specify a new user ID and password:

Procedure

1. Select **Query > Set User Information**.
The **Log on to Data Source** window opens.
2. In the **User name** field, type the user ID that you want QMF to use to connect to the repository or the data source.
3. In the **Password** field, specify the password associated with the user ID that you typed in the **User name** field.
4. Select the **Remember the password** check box to specify that QMF remember the password for the user ID that you entered in the **User name** field.
5. Select the **Use this information for subsequent data source connections** check box if you want QMF to use the specified User name and password for subsequent Data Source connections.
6. Click **OK** to set the user information that you have specified. QMF will use this user information the next time it connects to the repository or data source.

Changing user interface configuration information

The QMF for Workstation user application interface configuration information consists of the JDBC driver information and repository connection settings that are used to connect to shared repositories and data sources.

About this task

Advanced QMF for Workstation users can make changes to the configuration information from the user interface. Most users will never change the configuration information. The following topics describe how advanced QMF for Workstation users can change the user interface configuration information.

Setting up JDBC driver information

QMF uses JDBC to connect to all database repositories and data sources. QMF does not include JDBC drivers. The location of the JDBC drivers that QMF will use to connect to repositories and data sources must be defined.

About this task

QMF administrators are responsible for identifying where each instance of the application interface will find the JDBC drivers. Most users will never change the JDBC driver location information. Advanced users can change or add JDBC driver location information. The following topics describe how advanced users can change or add JDBC driver location configuration information.

Importing the JDBC library settings

You can use the JDBC library settings file to set up the JDBC driver information.

About this task

QMF administrators are responsible for distributing the JDBC driver information that you will need in order to access any data source. In most cases, administrators distribute this information by deploying a preconfigured JDBC library settings file when the application interface that you are using is installed.

Another method administrators can use to distribute this information is to create the JDBC library settings file and make it available to all users who will then import the file using their version of the application interface.

To import the JDBC library settings file:

Procedure

1. Select **File > Import > Connection and JDBC library settings**. The **Import Connection and JDBC Library Settings** window opens.
2. Specify where to find the `settings.xml/settings.zip` settings file in the **Settings file** field. You can use **Browse** to locate the file.
3. Make sure that the **JDBC library settings** check box is selected and the **Repository connections** check box is not selected.

Note: If the `settings.zip` archive includes the JAR driver files, and you want to import them as well, you must select the **Driver files** check box. Otherwise, make sure it is not selected.

4. Specify how you want conflicts with previous JDBC library settings to be resolved using **Conflict resolving**. Select **Override existing settings** to replace all conflicting information with new information. Select **Keep existing settings** to not update any existing information with new information. Select **Manually resolve conflicts** to open the **Conflicts** window where you will manually resolve conflicts between existing and new JDBC library settings.
5. Click **Finish**. The **Import Connections and JDBC Library Settings** window closes. The JDBC library information is saved. You can display your JDBC library information by selecting **Preferences** from the **View** menu and then select **JDBC Libraries**.

Related tasks

[Setting JDBC preferences](#)

You can use the **Preferences** window to set up the JDBC driver information.

Setting JDBC preferences

You can use the **Preferences** window to set up the JDBC driver information.

About this task

QMF administrators are responsible for distributing the JDBC driver information that users will need in order to access any data source. In most cases, administrators distribute this information by deploying a preconfigured JDBC library settings file when the application interface that you are using is installed.

Advanced users can define or change JDBC driver location information using the **JDBC Libraries** page of the **Preferences** window to specify the JDBC driver information for each type of database (such as Db2, Informix, Oracle) that will be accessed. The JDBC driver locations must be accessible to QMF. QMF will retain the JDBC driver information. You would only repeat this specification process to add new drivers or make changes to the properties of an existing driver.

To define JDBC driver information:

Procedure

1. Select **View > Preferences** to open the **Preferences** window. Select **JDBC Libraries**. The **JDBC Libraries** page opens. All of the JDBC driver libraries that you have already set up are listed in the **JDBC driver libraries** area. The JAR files associated with selected driver library are listed in the **Driver files** area. Additional information about the selected driver library is displayed on the **Details** panel.
2. To add new JDBC driver, select **Add Driver**. The **Add New Driver Library** window opens.
3. The JDBC driver information for each type of database is organized by libraries. Enter a name for the library in the **Enter new library name** field. It is recommended that the name that you specify describes the JDBC information that will be included in the library. For example, if you are adding JDBC driver information for accessing Db2 databases, you might call the library Db2. However, this is a descriptive field and can include any text.
4. Specify the actual name of the driver class that will be used in the **Driver class name** field. This can be found in your JDBC driver documentation. For example: com.ibm.db2.jcc.DB2Driver.
5. Optional: Specify a generic example of a correctly formatted URL that could be used to connect to the database in the **URL templates** field. For example, if you are adding JDBC driver information for accessing Db2 databases, you might specify the following JDBC URL template: jdbc:db2://host:50000/database. The generic information as specified in the template is presented when you are adding data sources to the repository and when you are creating repository connections. You will replace the generic information with the specific database information.
6. Click **OK** to create the driver library and close the **Add New Driver Library** window.
7. If you are using QMF for Workstation, add the location information for the JDBC driver files to the library. If you are using QMF for Workstation, click **Add Driver Files**. The **Add Files to [libraryname]** window opens. Search for and select the JDBC driver files that you want to add. Click **Open**. The location of the JDBC driver files is saved in the JDBC library that you have selected.
8. You can make the following changes to the JDBC driver information:
 - To edit the JDBC driver library information, select an existing JDBC driver library from the list and click **Edit Driver**. The **Edit Driver Library** window opens where you can make changes to the **Library name**, **Driver class name**, and **JDBC URL template** fields.
 - If you are using QMF for Workstation, to edit the name and location of the files that contain the JDBC drivers, select a file from the list of files that have been added to the JDBC library, and click **Edit**.
 - To remove a JDBC driver library, select an existing JDBC driver library from the list and click **Remove Driver**. The library, including all the JAR files that it contains, is removed.
 - If you are using QMF for Workstation, to remove a JAR file from a JDBC driver library, select an existing file from the list and click **Remove**. To remove a JAR file from a JDBC driver library, select an existing file from the list and click **Remove Driver File**. The file is removed.

9. Click **OK**. The JDBC library information is saved. The **Preferences** window closes. You must repeat this process for each type of JDBC driver that will be used to access various database data sources such as Informix or SQL Server.

Related tasks

[Importing the JDBC library settings](#)

You can use the JDBC library settings file to set up the JDBC driver information.

Setting up repository connection information

Advanced users can make changes to the repository connection information.

Most users will never change the repository connection information. The following topics describe how advanced users can change the repository connection configuration information.

Creating connections by importing the repository connections file

Advanced users can create repository connections by importing the repository connections file that is distributed by your QMF administrator.

About this task

QMF administrators are responsible for distributing the connection information that you will need in order to connect to a repository and access any data source. In most cases, administrators distribute this information by deploying a pre-configured JDBC library settings file when the application interface that you are using is installed.

Another method administrators can use to distribute this information is to create the repository connections settings file and make it available to all users who will then import the file using their version of QMF.

To import the repository connections settings file:

Procedure

1. Select **File > Import > Connection and JDBC library settings**.

The **Import Connection and JDBC Library Settings** window opens.

2. In the **Settings file** field, specify where to find the `settings.zip/settings.xml` file.

You can use **Browse** to locate the file.

3. Make sure that the **Repository connections** check box is selected and the **JDBC library settings** check box is not selected.

4. Specify how you want conflicts with previous repository connection settings to be resolved using **Conflict resolving**.

Select **Override existing settings** to replace all conflicting information with new information. Select **Keep existing settings** to not update any existing information with new information. Select **Manually resolve conflicts** to open the **Conflicts** window where you will manually resolve conflicts between existing and new JDBC library settings.

5. Click **Finish**.

The **Import Connections and JDBC Library Settings** window closes. The repository connections information is saved. The connection that you have just created will appear in the **Repository Connections** view. If you have QMF Registry configured for your QMF client, select **Local** from the **Server** list in the **Repository Connections** view. To connect to the repository, right-click the connection information for the repository to which you want to connect, and select **Connect**.

Related tasks

[Creating the repository connection for QMF](#)

QMF needs to know how to access the application data (as grouped by the repository name) so that users will have access to this data. The way to provide this connection data to QMF is to create it using the QMF connection wizard.

Managing repository connections

You manage your repository connections from the **Repository Connections** view. From this view you can edit and remove repository connections and you can connect and disconnect from a repository.

Creating the repository connection for QMF

QMF needs to know how to access the application data (as grouped by the repository name) so that users will have access to this data. The way to provide this connection data to QMF is to create it using the QMF connection wizard.

About this task

QMF administrators are responsible for distributing the connection information that users need to connect to a repository and access any data source. In most cases, administrators distribute this information by deploying a pre-configured repository connections settings file when the application interface is installed.

Advanced users can change or create repository connection information using the **Create New Repository Connection** wizard. The repository connection information is saved in a file called `settings.xml`. This file is saved in the personal settings directory of the machine that is running the application interface.

To create repository connection information:

Procedure

1. Open the **Create New Repository Connection** wizard by right-clicking anywhere in the **Repository Connections** view and selecting **New > Repository Connection**.
2. Select the type of connection that you will create from one of the following radio buttons:
 - **Shared repository** to create a repository connection to a shared repository using a client/server JDBC connection. With client/server access, the user's workstation establishes a direct connection to the repository database using its locally installed JDBC driver. For this type of connection each workstation that runs QMF for Workstation and the server (workstation) that runs QMF for WebSphere must have JDBC drivers installed.
 - **File-based repository** to create a repository connection to a file-based repository stored at a given location. (QMF for Workstation only)
 - **Network repository** to create a repository connection to a shared repository using direct network access. Network connectivity permits users to access a shared repository without the need for direct database connectivity from the workstation. Connecting to repositories using a network connection allows for consolidation of data access at a central server and removes the need to distribute JDBC drivers across each desktop running QMF for Workstation.
 - **Web service repository** to create a repository connection to a shared repository through the QMF for WebSphere web services API using a HTTP or HTTPS connection. Web service connectivity permits users to access a shared repository without the need for direct database connectivity from the workstation. Connecting to repositories using a web service connection allows for consolidation of data access at a central server and removes the need to distribute JDBC drivers across each desktop running QMF for Workstation.
3. Click **Next**. The second page of the **Create New Repository Connection** wizard opens. The second page of the wizard differs depending on the type of the repository connection that you are defining.

Related tasks

[Creating connections by importing the repository connections file](#)

Advanced users can create repository connections by importing the repository connections file that is distributed by your QMF administrator.

Managing repository connections

You manage your repository connections from the **Repository Connections** view. From this view you can edit and remove repository connections and you can connect and disconnect from a repository.

Creating shared repository connections

You create a shared repository connection to set up the information that will enable you to connect to a shared repository that resides in a database using a client/server JDBC connection.

About this task

The steps in this task describe how to create a shared repository connection.

Note: The *connection mode* for the shared repository connection is determined by settings on the **Protection** tab of the **Edit Repository Storage** window.

To complete the process of creating a shared repository connection:

Procedure

1. Open the second page of the **Create New Repository Connection** wizard.
2. Specify a name for the repository connection in the **Connection name** field. This name will be used to identify the connection as it appears in the **Repository Connections** view. You can enter any name but it is recommended that the name identifies the repository for which this connection information applies.
3. Select the **Connect immediately** check box to connect to the shared repository immediately. If this check box is clear, the connection information will be created, saved and displayed in the **Repository Connections** view, but you will not be immediately connected to the repository.
4. In the **JDBC Driver** field specify the library for the JDBC driver that will be used to access the database that is hosting the repository storage. You can select from a drop down list of libraries that you defined when you set up the JDBC driver information. You can use the ellipsis (...) to the right of the field to open the **JDBC Libraries** dialog where you can create additional JDBC driver libraries.
5. In the **JDBC URL** list, specify the URL that points to the database that hosts the repository storage within which the repository to which you want to connect resides.

The format for the URL can be displayed in the list.

Note: You can also use the **Build URL by URL Template** dialog box to form the correct string.

Click **Build URL** to open the **Build URL by URL Template** dialog box. From the JDBC URL drop-down list, select the template. In the table, specify the **server**, **port**, and database information and click **OK**. The result URL string is added to the JDBC URL list.

6. Click **Advanced** to specify any driver-specific connection string keywords and their values that will be used when connecting to this shared repository. For information on any keywords that are required by a selected database driver, consult the specific driver's documentation.

Note: For the DB2® Type 4 UDB driver, to receive extended SQL exception error messages, check the **retrieveMessagesFromServerOnGetMessage** option in the **Advanced JDBC Settings** window.

7. To configure a PassTickets-based Single Sign-on connection between QMF and Db2 for z/OS database, click **Single Sign-on**.
8. In the **JDBC Driver** field specify the library for the JDBC driver that will be used to access the database that hosts the repository. Select from a drop-down list of libraries that you defined when you set up the JDBC driver information. Click the ellipsis (...) to open the **JDBC Libraries** window where you can create additional JDBC driver libraries.
9. In the **JDBC URL** field specify the URL that points to the database that hosts the repository storage within which the repository to which you want to connect resides. The format for the URL can be displayed in the field if you specified an example when setting up the JDBC libraries.

10. Optional: If you are using the Direct DRDA Connection to connect to the repository for the first time, you must bind Direct DRDA packages. To do so, complete the following steps:
 - a) Click **Bind Packages**.
 - b) In the **Owner ID** field, enter a privileged SQLID with the necessary rights for the binding process. If the value of this field is not specified, the user name from the **Set User Information** dialog will be used.
 - c) Click **Install**.
11. Click **Set User Information** to specify the user information that will be used to connect to the repository. The **Set User Information** window opens.
 - a) Select the **User ID and password are required** check box to specify that user information will be requested each time a connection is made to this shared repository. In some cases, leaving this check box clear will enable users to access the repository without overtly specifying their user information.
 - b) Enter the user ID and password that will be used to access the specified repository database in the **User name** and **Password** fields.
 - c) To remember the password, select the **Remember password** check box.

If you do not select the **Remember password** check box, the system will prompt you for a user ID and password whenever you open the **Edit Repository Connection** wizard.
 - d) To enable the *Single-Use Password Mode*, click **Advanced** and select **Enabled**.

In the **Repository storage connections timeout** field, enter the time in minutes that the Single-Use Password Mode feature will remain enabled (while not being used) without being closed by application.

Note: A value of zero 0 sets unlimited idle state for the connection timeout, meaning that connection will close only when the session ends.

Enabling the *Single-Use Password Mode* feature provides users with the ability to log into the repository for the entire session without having to reenter his or her user ID and password.
12. Click **OK** to close the **User Information** window.
13. Select **Hide from web service clients** to exclude this connection from the list of available connections in the **Repository connection name** field in QMF for WebSphere.

This check box is displayed in QMF for WebSphere user interface only. If you are working in the QMF for Workstation user interface, **Hide from web service clients** is not an option.

If you select this check box, the user cannot establish a web service connection to this repository connection.
14. In the **Repository name** field, select the repository to connect to. If no repositories are listed, click **Refresh**.
15. Click **Finish**.

The repository connection that you have just created will appear in the **Repository Connections** view. If you have QMF Registry configured for your QMF client, select **Local** from the **Server** list in the **Repository Connections** view.

If you selected the **Connect immediately** check box, you are connected to this repository. This repository connection information is entered in the connections settings file.

Related tasks

Creating file-based repository connections

If you are using QMF for Workstation, you can create a file repository connection to set up the information that will enable you to connect to a file-based repository that is stored on a local or network drive.

Creating network repository connections

You create a network repository connection to set up the information that will enable you to connect to a shared repository using direct network access.

Creating web service repository connections

You create a Web service repository connection to set up the information that will enable you to connect to a shared repository through the QMF for WebSphere web services API using a HTTP or HTTPS connection.

Creating file-based repository connections

If you are using QMF for Workstation, you can create a file repository connection to set up the information that will enable you to connect to a file-based repository that is stored on a local or network drive.

About this task

File connections can only be created for legacy, file-based repositories. The later versions of QMF for Workstation and QMF for WebSphere have replaced file-based repositories with personal repositories.

To complete the creation of a file-based repository connection:

Procedure

1. Open the second page of the **Create New Repository Connection** wizard.
2. Specify a name for the repository connection in the **Connection name** field. This name will be used to identify the connection as it appears in the **Repository Connections** view. You can enter any name but it is recommended that the name identifies the repository for which this connection information applies.
3. Select the **Connect immediately** check box to connect to the file-based repository immediately. If this box is clear, the connection information will be created, saved and displayed in the **Repository Connections** view, but you will not be immediately connected to the repository.
4. Click **Finish**.

The repository connection that you have just created will appear in the **Repository Connections** view. If you have QMF Registry configured for your QMF client, select **Local** from the **Server** list in the **Repository Connections** view.

If you selected the **Connect immediately** check box, you are connected to this repository. This repository connection information is entered in the connections settings file.

Related tasks

Creating shared repository connections

You create a shared repository connection to set up the information that will enable you to connect to a shared repository that resides in a database using a client/server JDBC connection.

Creating network repository connections

You create a network repository connection to set up the information that will enable you to connect to a shared repository using direct network access.

Creating web service repository connections

You create a Web service repository connection to set up the information that will enable you to connect to a shared repository through the QMF for WebSphere web services API using a HTTP or HTTPS connection.

Creating network repository connections

You create a network repository connection to set up the information that will enable you to connect to a shared repository using direct network access.

About this task

Network connectivity permits users to access a shared repository without the need for direct database connectivity from the workstation. Connecting to repositories using a network connection allows for

consolidation of data access at a central server and removes the need to distribute JDBC drivers across each desktop running QMF for Workstation.

QMF for Workstation users can connect to repositories via a centralized machine running the QMF network repository service. Network connectivity provides similar capabilities to the web-based approach but does not require a web application server in order to operate. In order to use network connectivity you must set up and start the network repository service.

Note: When you create a network repository connection in QMF for Workstation (client) the version of network repository service (server) must be at the same release levels. If the release levels (between the client and server) do not match, network repository connections will not work.

To complete the creation of a network repository connection:

Procedure

1. Verify with the QMF administrator that the network repository service has been set up and started.
2. Open the second page of the **Create New Repository Connection** wizard.
3. Specify a name for the repository connection in the **Connection name** field. This name will be used to identify the connection as it appears in the **Repository Connections** view. You can enter any name but it is recommended that the name identifies the repository for which this connection information applies.
4. Select the **Connect immediately** check box to connect to the repository immediately. If this check box is clear, the connection information will be created, saved and displayed in the **Repository Connections** view, but you will not be immediately connected to the repository.
5. Specify the name or the IP address of the server (or workstation) that is running your network service in the **Server host** field.
6. Specify the contact port for the network server in the **Port** field.
7. Click **Refresh**.

The available repository connections for the specified network repository service are retrieved. Select the repository connection that you want to use from the list of available repository connections.

8. Click **Finish**.

The repository connection that you have just created will appear in the **Repository Connections** view. If you have QMF Registry configured for your QMF client, select **Local** from the **Server** list in the **Repository Connections** view.

If you selected the **Connect immediately** check box, you are connected to this repository. This repository connection information is entered in the connections settings file.

Related tasks

[Creating shared repository connections](#)

You create a shared repository connection to set up the information that will enable you to connect to a shared repository that resides in a database using a client/server JDBC connection.

[Creating file-based repository connections](#)

If you are using QMF for Workstation, you can create a file repository connection to set up the information that will enable you to connect to a file-based repository that is stored on a local or network drive.

[Creating web service repository connections](#)

You create a Web service repository connection to set up the information that will enable you to connect to a shared repository through the QMF for WebSphere web services API using a HTTP or HTTPS connection.

Creating web service repository connections

You create a Web service repository connection to set up the information that will enable you to connect to a shared repository through the QMF for WebSphere web services API using a HTTP or HTTPS connection.

About this task

You can use web-based connectivity to access a shared repository without the need for direct database connectivity from the workstation. By using this method to connect to repositories you consolidate data access at a central server, thus removing the need to distribute JDBC drivers across each desktop running QMF for Workstation.

Web-based connectivity is provided by the QMF for WebSphere built-in repository web service. To enable this facility, you must have QMF for WebSphere deployed to a WebSphere Application Server (WAS) instance. In addition, you must have repository connection information available to this version of QMF for WebSphere. QMF for Workstation users connect to the repository using the URL and port to the QMF for WebSphere server.

Note: When you create a web repository connection in QMF for Workstation (client) the version of web repository service (server) must be at the same release levels. If the release levels (between the client and server) do not match, web repository connections will not work.

To complete the creation of a web service repository connection:

Procedure

1. Ensure that QMF for WebSphere has been deployed to a WebSphere Application Server (WAS) instance by your administrator.
2. Open the second page of the **Create New repository Connection** wizard.
3. Specify a name for the repository connection in the **Connection name** field. This name will be used to identify the connection as it appears in the **Repository Connections** view. You can enter any name but it is recommended that the name identifies the repository for which this connection information applies.
4. Select the **Connect immediately** check box to connect to the web service repository immediately. If this box is clear, the connection information will be created, saved and displayed in the **Repository Connections** view, but you will not be immediately connected to the repository.
5. In the **Web Server URL** field, specify the URL to launch the instance of QMF for WebSphere whose web services API will be used for this connection. This is the URL address that was created when the instance of QMF for WebSphere was deployed.
6. Click **Refresh**. The available repository connections that have been defined for the instance of QMF for WebSphere that you specified in the **Web Server URL** field are retrieved. Select the repository connection that you want to use from the list of available connections.
7. Click **Finish**.

The repository connection that you have just created will appear in the **Repository Connections** view. If you have QMF Registry configured for your QMF client, select **Local** from the **Server** list in the **Repository Connections** view.

If you selected the **Connect immediately** check box, you are connected to this repository. This repository connection information is entered in the connections settings file.

Results

Currently, when the QMF for WebSphere client opens, it connects to the last accessed repository. An additional **connection** URL parameter has been added to control which repository connection should be used when the QMF for WebSphere client opens.

For example:

```
http://host:port/QMF/user?repository=MyConnection
```

If the **connection** parameter points to an invalid connection, then the system opens open the **Switch Repository** window. You can pass logon information for secured repository connections using the **user** and **password** parameters.

For example:

```
http://host:port/QMF/  
user?repository=MyConnection&user=userLogin&password=userPassword
```

Related tasks

[Creating shared repository connections](#)

You create a shared repository connection to set up the information that will enable you to connect to a shared repository that resides in a database using a client/server JDBC connection.

[Creating file-based repository connections](#)

If you are using QMF for Workstation, you can create a file repository connection to set up the information that will enable you to connect to a file-based repository that is stored on a local or network drive.

[Creating network repository connections](#)

You create a network repository connection to set up the information that will enable you to connect to a shared repository using direct network access.

Managing repository connections

You manage your repository connections from the **Repository Connections** view. From this view you can edit and remove repository connections and you can connect and disconnect from a repository.

About this task

Each QMF user (with and without administrator privileges) manages their own repository connections. Repository connection information is stored in a user's home directory by operating system user ID. Each user can add, edit, and delete repository connections from their home directory.

To add, connect to, edit, or delete a repository connection:

Procedure

1. Open the **Repository Connections** view. From the **Server** list, select the QMF Server whose repository connections you want to display, or select **Local** to display the repository connections that are available on your local machine. Local repository connections can be created by users or supplied to users by the administrator.

If the QMF Server that you have selected from the **Server** list has one or more registered QMF Catalogs, they will be available as a separate connection in the **Repository Connections** view. If you establish this connection, you will be able to access QMF Catalog objects from the **Repository Explorer** view.

Note: QMF Server-based repository connections will be available to you only if you have an instance of QMF configured for your QMF client.

2. To edit a repository connection's property information, complete the following steps:
 - a) Right-click the repository connection that you want to edit and select **Edit** from the pop-up menu.

The **Edit Repository Connection** wizard presents edit dialogs based on the type of connection that you selected.

b) From these dialogs make changes to the repository connection information.

Upon completion, the modified connection information is saved.

3. To remove a repository connection, complete the following steps:

a) Right-click the repository connection that you want to remove and select **Remove connection** from the pop-up menu.

A message asking you to confirm the removal is issued.

b) Click **OK** to remove the repository connection.

4. To connect to a repository, right-click the repository connection in the list and select **Connect** from the pop-up menu.

You are immediately connected to the repository that you selected.

5. To disconnect from a repository, right-click the repository connection from which you want to disconnect and select **Disconnect** from the pop-up menu.

You are immediately disconnected from the repository that you selected.

6. To reconnect to the current repository using a different user ID, right-click the current repository connection and select **Reconnect** from the pop-up menu.

A **Logon** window opens where you can specify a different user ID and password that will be used to connect to the current repository.

7. Click **OK** and you are reconnected to the same repository but with a different user ID.

Related tasks

[Creating connections by importing the repository connections file](#)

Advanced users can create repository connections by importing the repository connections file that is distributed by your QMF administrator.

[Creating the repository connection for QMF](#)

QMF needs to know how to access the application data (as grouped by the repository name) so that users will have access to this data. The way to provide this connection data to QMF is to create it using the QMF connection wizard.

Configuring repository caching

You can configure a caching functionality on your repositories. When caching is enabled on a repository, the system saves the content of the repository (dashboards, queries, visual reports, and other objects) to the memory on the workstation client or on the web server if you are using QMF for WebSphere.

About this task

This task describes how to configure repository caching.

Considerations:

- Caching happens in local memory on your machine if you are using QMF for Workstation and in memory on the web server if you are using QMF for WebSphere.

Subsequently, the next time you access the repository objects, information is retrieved from memory (without sending a request to the repository), hence retrieval time is reduced.

The following content is cached:

- Content of objects
- Path to objects
- Metadata to tables
- When you make changes to cached repository objects, the cache is refreshed to reflect the changes.

- For QMF for Workstation, the repository cache is cleared whenever you shut down your machine or close the application.
- For QMF for Workstation, users can set repository caching parameters for the version of the application running on their machine.
- For QMF for WebSphere, repository caching is shared between all connections made with the same Repository Connection login.
- For QMF for WebSphere, you must have administrator authorization in order to change repository connection settings.

To configure repository caching:

Procedure

1. Open the second page of the **Creating New Repository Connection** wizard or of the **Edit Repository Connection** wizard.

2. In the **Repository cache** area, select the **Enable** check box to activate cache configuration options.

The radio buttons for predefined cache validation are enabled in the **Predefined modes** section.

To implement a predefined cache validation mode, select the associated radio button. The cache validation processing for each predefined mode are as follows:

- **Low relevance-High performance**

If selected, cache validation is not performed.

- **Balanced relevance/performance**

If selected cache validation is performed every 5 minutes. This is the default setting for cache validation.

- **High relevance-Low performance**

If selected cache validation is performed every 60 seconds (1 minute).

3. To implement custom cache validation, select the **Use custom cache settings** check box.

When you select the **Use custom cache settings** check box, the options for predefined cache validation modes are disabled.

Click **Custom. . .** to launch the **Repository Cache Settings** wizard.

Use this wizard to specify cache size information and to set validation intervals.

The wizard consists of two panes:

- The **Repository Caches** pane on the left

This pane lists the specific types of repository-related data to cache.

- The **Repository Caches Settings** pane on the right

This pane is where you set the caching parameters.

You can set parameter values for general caching, which are applied to each type of repository-related data.

You can also set parameter values for the different types of repository-related data. These types include:

- **Content Cache**
- **Server Metadata Cache**
- **Structure Cache**
- **Table Metadata Cache**

- a) To set general cache settings, click **Repository Caches** in the list pane and set a value in the **Cache validation time (minutes)** field.

The value you enter represents how often (in minutes) that cache validation occurs.

The value you enter determines how often the cache validates that the objects in it haven't changed. The system checks the last updated date for every cached object stored in the repository.

The default value is 5 minutes.

Any object that has been changed gets removed from the cache.

Any object that has not been accessed within the time specified, are not removed from the cache.

Enter 0 (zero) if you never want to refresh the cache. This is useful for repositories with static content.

- b) To set parameter values for the different types of repository-related data, expand **Repository Caches** and select the type of data to configure.

The types of repository-related data and their configuration parameters are as follows:

- **Content Cache**

Includes the following parameter:

- **Content size (megabytes)** specifies the maximum size allowed (in megabytes) for cached content.

The value entered represents the total combined size of all objects stored in the cache and saved to memory on your machine if you are using QMF for Workstation and on the web server if you are using QMF for WebSphere.

When a new object is added to the cache whose size has reached the maximum allowed size, the system removes objects until it has enough space to accommodate the new object.

Those objects that have not been accessed for the longest period of time are removed first.

- **Server Metadata Cache**

Includes the following parameters:

- **Lifetime limit (minutes)**

Parameter sets maximum time (in minutes) for records to be kept in cache. Lifetime is calculated for each record starting from the moment of its creation. Upon expiration of specified lifetime record is forced out of cache.

- **Inactivity (minutes)**

Parameter sets maximum time (in minutes) for records to be kept in cache without being referenced. Inactivity period is calculated for each record starting from the moment of its last use. Upon expiration of specified inactivity period record is forced out of cache.

- **Structure Cache**

Includes the following parameters:

- **Lifetime limit (minutes)**

Parameter sets maximum time (in minutes) for records to be kept in cache. Lifetime is calculated for each record starting from the moment of its creation. Upon expiration of specified lifetime record is forced out of cache.

- **Inactivity (minutes)**

Parameter sets maximum time (in minutes) for records to be kept in cache without being referenced. Inactivity period is calculated for each record starting from the moment of its last use. Upon expiration of specified inactivity period record is forced out of cache.

- **Table Metadata Cache**

Includes the following parameters:

- **Lifetime limit (minutes)**

Parameter sets maximum time (in minutes) for records to be kept in cache. Lifetime is calculated for each record starting from the moment of its creation. Upon expiration of specified lifetime record is forced out of cache.

– **Inactivity (minutes)**

Parameter sets maximum time (in minutes) for records to be kept in cache without being referenced. Inactivity period is calculated for each record starting from the moment of its last use. Upon expiration of specified inactivity period record is forced out of cache.

4. Click **Finish** to exit the wizard.

Results

You have enabled repository caching and have set the parameters by which caching occurs. The settings are saved to the product configuration.

The system allows you to read and write (save) file Cache settings and uses it to set the current settings directly to the Caches. All settings are grouped by services in the set of configuration records.

Importing Settings from QMF for Windows

You can import QMF for Windows settings and preferences.

About this task

By importing settings and preferences you eliminate the need to recreate them manually when you configure QMF for Workstation.

The Import QMF for Windows Configuration Settings wizard opens automatically the first time you launch QMF for Workstation. You can also open the wizard by selecting **File > Import > QMF for Windows Configuration Settings**.

In the wizard, you can select the settings and user preferences that you want to import.

QMF for Windows settings

QMF for Windows settings can be imported to QMF for Workstation.

The following table lists the QMF for Windows settings that you can import by using the Import QMF for Windows Configuration Settings wizard.

| Table 5. The settings that you can import from QMF for Windows | | | |
|--|---|---|---|
| Setting | Description | To view in QMF for Windows | To view in QMF for Workstation |
| Favorites | The contents of the <i>Favorites</i> folder | Open Database Explorer tree | Open Personal view |
| Recently used objects | The list of recently used objects | Open Database Explorer tree | Open Personal view and Repository Explorer view |
| Global variables | The list of global variables | Select View > Global Variables | Select View > Preferences > Global variables |
| Saved join definitions | The list of joins that are added automatically on the basis of information about the tables that you have ever joined in Prompted or Query Builder editors. | | |
| Export preferences | The settings for exporting query result and export report type to a file or database. | Select Results > Save to File or Results > Save to Database | Select File > Export |
| LOB options | LOB object settings | Select View > Options > LOBs | Select View > Preferences > LOBs |

| Setting | Description | To view in QMF for Windows | To view in QMF for Workstation |
|-------------------------|--|--|--|
| Grid options | The default font properties, formatting options, cell colors and other grid settings | Select Results > Font and Results > Format . | Select Results > Font and Results > Format |
| Page setup settings | The page settings that are used for printing | Select File > Page Setup | Select File > Page Setup |
| User interface settings | The user interface settings such as language, default font and null values display options | Select View > Options > General and View > Options > Appearance | Select View > Preferences > General and View > Preferences > Appearance |
| Command bar history | The history of commands entered in command bar | | |

Correct format for identifiers

The maximum length of table owner, table name, object owner, object name, as well as column name identifiers depends on the version and platform of Db2.

When specifying the identifier you can use any normal character. Normal characters include uppercase letters, digits, or the following characters: `_`, `@`, `#`, and `$`. You must enclose any special characters in quotes. If the identifier includes a quote character, you double each occurrence of the quote character and then enclose the entire identifier in quote characters. For example, you enter the identifier `pro"ject` as `"pro"ject`. If you use lowercase letters and do not enclose the identifier in quotes, any lowercase letters are changed to uppercase.

Matching patterns

When entering query, form and table names for searching, you can use the percent (%) and underscore (_) characters to match patterns.

You use the percent character (%) to match a string of any length containing any characters. For example, to list all items beginning with the letter A, you enter `A%`. To list all items, enter just the % character.

You use the underscore character (_) to match a single character. For example, to list all items that have the letter A in the second position, enter `_A%`.

If the pattern you enter contains special characters, you must enclose the entire pattern in quotation marks. For example, to include a space as part of a pattern, you enter `"A B%"`. Special characters include any characters other than:

- A through Z (uppercase only)
- 0 through 9
- #, \$, @, and _

Creating new objects or repository items

QMF supplies numerous wizards to help you create new objects.

About this task

From the **New** window you can select the type of object that you want to create and launch the wizard that will help you create that object.

To launch the new object wizard:

Procedure

1. To open the **New** window, select **File > New > Other**.

The **New** window opens.

The wizards that are available are listed in the **Wizards** list box.

2. From the

Environments folder, you can select one of the following wizards:

- **Data Source Mapping** to add a data source to the environment.
- **Environment** to create an environment.

3. From the **QMF Objects** folder, you can select one of the following wizards:

- **Ad hoc Report** to create an Ad hoc report.
- **Analytical Query** to create an analytical query.
- **Drill-down Path** to create a drill-down path.
- **ER Diagram** to create an Entity Relationship (ER) diagram. The **ER Diagram** wizard is available only if you have the Administrator component installed.
- **ER Diagrams Folder** to add an ER Diagram folder to the data source. The **ER Diagrams Folder** wizard is available only if you have the Administrator component installed.
- **Forecast** to create a forecast.
- **Form** to create a form that can be used to generate a classic report.
- **JavaScript Data Source Schedule** to create a JavaScript data source schedule.
- **JavaScript Module** to create a JavaScript module and open the JavaScript code editor.
- **JavaScript Table** to create a JavaScript table.
- **OLAP Query** to create a OLAP query.
- **Procedure** to create a procedure.
- **Prompt Hierarchy** to create a prompt hierarchy.
- **Query using the Draw Query wizard** to create a query using the **Draw Query** wizard.
- **Quick Report** to create a quick report.
- **Virtual Data Source Schedule** to specify cache expiration settings for a virtual data source.
- **Virtual Table** to create a virtual table.
- **Visual Dashboard** to create a visual dashboard project. (QMF for Workstation only)
- **Visual Dashboard Template Category** to create a visual dashboard template category. (QMF for Workstation only)
- **Visual Query** to create a visual (SQL) query.
- **Visual Report** to create a visual report project. (QMF for Workstation only)
- **Visual Report Template Category** to create a visual report template category. (QMF for Workstation only)

4. From the **Repository** folder, you can select one of the following wizards:

- **Folder** to add a folder to your repository or workspace.
- **Folder Data Source** to create a folder data source.
- **Link** to create a shortcut to an object.
- **Multidimensional Data Source** to create a multidimensional data source.
- **Personal Repository** to create a new personal repository.
- **Repository Connection** to create a new repository connection.
- **Workspace** to create a new repository workspace.

5. The following wizards are available from the **Repository** folder only if you have the Administrator component installed:

- **Hive Data Source** to create a Hive data source.

- **JavaScript Data Source** to create a JavaScript data source.
 - **Relational Data Source** to add a relational data source to your repository.
 - **Virtual Data Source** to add a virtual data source to your repository.
 - **Shared Repository** to create a shared repository.
 - **Shared Repository Storage** to create a shared repository storage.
6. From the **Other** folder, you can choose one of the following wizards:
- **Scheduled Task** to create a scheduled task.
 - **Web Link** to create short URLs to QMF objects. You can use these URLs to open the objects directly in web browsers. The **Web Link** wizard is available only when there is an active web service repository connection.
7. Click **Next**.

The first page of the wizard that you selected opens.

The first page will vary depending on the object or repository item that you are creating.

Creating web links to data objects

Use the **Web Link** wizard to create web links to QMF objects. You can use these links to open the objects directly in web browsers.

Before you begin

If you are using QMF for Workstation, the **Web Link** wizard is available only when you have an available web service repository connection.

Note: QMF users can also create web links. However, they can view, edit, or delete only the web links that they created. As administrator, you can view, edit, or delete all the web links.

About this task

To create a web link to data objects:

Procedure

1. From the main menu, select **File > New > Other > Web Link** to open the **Web Link** wizard.

Note: You can also access this wizard from the repository tree pop-up menu option **Create Web Link**. If you access the wizard by using this method, the **Repository object** field and **User-Defined Parameters** are pre-filled automatically.

2. From the **Repository object** field, select the object to which you want to create a web link.
3. Specify settings for the following **Predefined Parameters**:

- The **Toolbar** check box indicates whether to show or hide the toolbar.
- The **Run** check box indicates whether to run the object when it is opened.
- The **Environment** check box indicates whether to use the default environment to run the object.

If you want to use an environment other than the default one, select the check box and choose from the list of environments.

- The **Renderer mode** check box is available for visual reports. It determines the mode in which visual report is rendered.

4. Specify **User-Defined Parameters** to represent the selected parameters of repository objects.

You can add parameters into the generated URL by selecting **Include in URL** check box.

When the **Toolbar** check box is selected, the URL formula is:

```
http://{host}:{port}/{app_context}/g?object_name&param1=value1
```


When the **Toolbar** check box is clear, for visual dashboards the URL formula is:

```
http://{host}:{port}/{app_context}/go?object_name&param1=value1
```

5. Click **Next** to display the second page of the **Web Link** wizard.

On the page, you can add data sources. Use the **Add Data Source** and **Remove Data Source** buttons to edit the list of data sources. In the **Login** and **Password** fields, you can enter logins and passwords for added data sources.

You can specify the access mode for the web link as **Public** or **Private**. Web links are private by default.

- For private web links, you can enter the data source credentials.
 - If you enter the data source credentials on this page, you need to provide only the repository storage credentials and repository credentials when you access the link.
 - If you do not enter the data source credentials on this page, you need to provide data source credentials, repository storage credentials, and repository credentials when you access the link.
- For public web links, you can enter the data source credentials, repository storage credentials, and repository credentials. If you enter these credentials on this page, you will not be prompted to enter them when you access the web link. The credentials are saved and used at the time of access.

In the **Web link name** field, enter the name for the current object, that you want to use in the URL and in the tab title in your browser.

For example, if you specify **NEW_OBJECT_NAME** in the **Web link name** field and check the **Toolbar** check box, the following URL is generated:

```
http://{host}:{port}/{app_context}/g?NEW_OBJECT_NAME
```

When the **Toolbar** check box is clear, the URL is:

```
http://{host}:{port}/{app_context}/go?NEW_OBJECT_NAME
```

You can change the link name by using the **Web link name** field.

If you leave the **Web link name** field empty, a new name is created automatically.

6. Click **Create** to generate a web link.

If the specified link name exists, you must either rename it, or leave the **Web link name** field empty to create the name automatically.

7. Click **Open Web Link** to open the created web link in the browser.

8. Click **Finish** to close the **Web Link** wizard.

Saving objects

You can save the objects that you create to a repository, to a QMF catalog, or to a file. In order to facilitate saving objects, QMF offers several save commands and save wizards.

The save commands are **Save**, **Save At** and **Save All**. The status of the current active object determines which save commands are available. The active object status can be one of the following:

- The object is new and has never been saved.
- The object is a previously saved object and no changes have been made to the open object.
- The object is a previously saved object and changes have been made to the open object.

The save commands perform as follows:

- If the object is a previously saved object and no changes have been made to the open object there are no save commands available.

- If the object is a previously saved object and changes have been made to the open object, you can use any of the save commands. The **Save** command will save the object in its original location. The **Save At** command will open the save wizards where you can save the object to a new location. The **Save All** command will sequentially save all active objects opening the save wizards only when the object is new and has never been saved.
- If the object is new and has never been saved, you can use any of the save commands to open the save wizards.

Saving objects to a file

You can save any objects (forms, queries, procedures) that you create to a file.

About this task

To save an object to a file:

Procedure

1. If you are using QMF for Workstation:
 - a. Select **Save At > File** from the **File** menu.
 - b. Navigate to the directory where you will save the object.
 - c. Specify the name of the object in the **File name** field.
 - d. Click **Save**. The object is saved to the specified location.
2. If you are using QMF for WebSphere:
 - a. Select **Save At > File** from the **File** menu.
 - b. Specify the name of the object in the **Input file name** field.
 - c. Click the **Link to file** hyperlink. A browser specific download dialog opens.
 - d. Download the object to the required location on your local machine.

Related tasks

[Saving objects in a repository or a repository workspace](#)

You can save the objects (forms, queries, procedures) that you create in a repository or in a particular repository workspace using the **Save at Repository** dialog.

[Saving objects in the QMF catalog](#)

You can save any objects (forms, visual queries, procedures) that you create to a QMF catalog if QMF catalog functionality has been enabled for the repository or workspace to which you are connected.

Saving objects in a repository or a repository workspace

You can save the objects (forms, queries, procedures) that you create in a repository or in a particular repository workspace using the **Save at Repository** dialog.

About this task

To save an object in a repository or a repository workspace:

Procedure

1. Select **Save At > Repository** from the **File** menu.
The **Save at Repository** window opens.
2. Each repository and repository workspace to which you have access is listed in the **Select parent entry** tree. Select where you want to save the object by selecting the parent node in tree structure.
3. Type the name that you want to identify the object in the **Name** field. The name field always displays the current name of the active object.

4. Type any comment in the **Comment** field. Comments are displayed when you view the properties of the object.
5. Click **OK**. The object is saved in a repository or a repository workspace.

Related tasks

Saving objects to a file

You can save any objects (forms, queries, procedures) that you create to a file.

Saving objects in the QMF catalog

You can save any objects (forms, visual queries, procedures) that you create to a QMF catalog if QMF catalog functionality has been enabled for the repository or workspace to which you are connected.

Saving objects in the QMF catalog

You can save any objects (forms, visual queries, procedures) that you create to a QMF catalog if QMF catalog functionality has been enabled for the repository or workspace to which you are connected.

About this task

To save an object to a QMF catalog:

Procedure

1. To open the **Save to QMF Catalog** wizard, select any of the **Save** commands from the **File** menu or click the **Save** toolbar button. The **Save [objectname]** window opens. Select **Save to QMF Catalog**. Click **Next**. The **Save to QMF catalog** window opens. If you are not connected to the QMF catalog data source, you will be asked for logon information.
2. Specify the name of the data source where the QMF catalog resides in the **Data Source** field.
3. Specify the owner ID of the object in the **Owner** field. You must specify the owner in the correct format for identifiers.
4. Specify a name for the object in the **Name** field. You must specify the name in the correct format for identifiers.
5. Specify a folder, which stores a link to the saved object, in the **Folder** field. Use folders to group objects in a convenient way.
6. Select the type of object that you are saving from the **Type** list. The types of objects that you can save in the QMF catalog are visual queries, forms, and procedures.
7. You can specify a comment in the **Comment** field.
8. Select **Share the object with other users** if you want to specify that the object is to be shared with other users. Do not select this check box if you want to specify that only you can use the object.
9. Click **Finish**. The object is saved in the QMF catalog.

Related tasks

Saving objects to a file

You can save any objects (forms, queries, procedures) that you create to a file.

Saving objects in a repository or a repository workspace

You can save the objects (forms, queries, procedures) that you create in a repository or in a particular repository workspace using the **Save at Repository** dialog.

Opening saved objects

The information that you must specify in order to open a saved object varies depending on where the object was saved.

About this task

Objects can be saved in a repository, in the QMF catalog, or in a file.

To open a saved object:

Procedure

1. Select **File > Open From**.
2. From the pop-up menu, select one of the following depending on where the object is saved.
 - **QMF Catalog** if the object that you want to open resides in a QMF catalog.
 - **Repository** if the object that you want to open resides in a repository.
 - **File** if the object that you want to open resides in a file.
3. The **Open From** window opens. The information that is requested depends on your selection.

Opening QMF catalog objects

You must specify the data source where the QMF catalog resides, and the object owner and name when you open a QMF catalog object.

About this task

You can open QMF objects in one of two ways:

Procedure

1. In the **Workspaces** or **Repository Explorer** view, open the QMF catalog that contains the object that you want to open, navigate to and double-click the object.
The QMF object opens in the editor.
2. To search for and open a QMF catalog using a window:
 - a) Select **File > Open From > QMF Catalog**.
 - b) In the **Data Source** field, specify the name of the data source where the QMF catalog resides.
 - c) In the **Owner** field, specify the owner of the object that you want to open. You must specify the owner in the correct format for identifiers.
 - d) In the **Name** field, specify the name of the object that you want to open. You must specify the name in the correct format for identifiers.
 - e) In the **Folder** field, specify the name of the folder that stores the object that you want to open. Click the ellipsis (...) to specify any other folder on this server.
 - f) To list the objects that have been saved in the QMF catalog on the specified data source, click the **From List** button. The **Object List** window opens where you search for, list and select an object to open.
 - g) Click **OK**. The object opens in the editor window.

Related tasks

Opening repository objects

To open a repository object you select from a list of objects that are available in the repository to which you are currently connected.

Opening objects saved to a file

To open an object that has been saved to a file, specify the name and location of the file that contains the object .

Opening repository objects

To open a repository object you select from a list of objects that are available in the repository to which you are currently connected.

About this task

To open an object that has been saved in a repository:

Procedure

1. Select **File > Open From > Repository**.
2. The current repository is displayed in a tree structure in the **Select repository entry(s)** list box. Expand the folders until you uncover the object that you want to open. Select the object.
3. Click **OK**. The object opens in the editor window.

Related tasks

Opening QMF catalog objects

You must specify the data source where the QMF catalog resides, and the object owner and name when you open a QMF catalog object.

Opening objects saved to a file

To open an object that has been saved to a file, specify the name and location of the file that contains the object .

Opening objects saved to a file

To open an object that has been saved to a file, specify the name and location of the file that contains the object .

About this task

To open an object that has been saved in a file:

Procedure

1. Select **File > Open From > File**.
2. An operating system specific **Open** window opens. Specify the name of the file and where it resides.
3. Click **Open**. The object opens in the editor window.

Related tasks

Opening QMF catalog objects

You must specify the data source where the QMF catalog resides, and the object owner and name when you open a QMF catalog object.

Opening repository objects

To open a repository object you select from a list of objects that are available in the repository to which you are currently connected.

Chapter 4. Creating queries

A query is a request for information from a data source. To request information from a relational data source you can construct a query in a standard fashion by using Structured Query Language (SQL) statements. Or, if you are not familiar with SQL, you can construct a query graphically by using the **Query Builder** editor.

Query types

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

The following table lists the different query types supported in QMF and provides usage information about each type.

| Type | Description | Characteristics of this type of query |
|------------------|---|---|
| Visual Query | A query written in SQL or created with the Query Builder editor. | <p>The Visual Query contains SQL and a set of formatting commands for QMF for Workstation and QMF for WebSphere family to execute on the result set:</p> <ul style="list-style-type: none">• grouping and aggregation of columns• formatting of the data cells and text• creation of Calculated Columns using a new more powerful syntax that includes user defined functions in JavaScript• instructions to build Charts that can be toggled through by the user• filter criteria that act on the result set• multilevel prompts for substitution variables <p>Results can be Exported or used with forms, Visual Reports, and Visual Dashboards.</p> |
| Analytical Query | Manipulates results sets of other queries to perform client side joins, appends, crosstabs and other types of analytical transformations. | <p>Saved in the Repository or as a file, NOT saved to the QMF Catalog.</p> <p>Not shared with QMF for TSO and CICS.</p> <p>Contains the SQL from all other queries referenced by it, and a set of formatting commands, identical to those available in the Visual Query, QMF for Workstation and QMF for WebSphere to execute on the result set.</p> <p>Results can be Exported or used with Forms, Visual Reports, and Visual Dashboards.</p> |

Related concepts

Query interfaces

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

Creating visual queries using the SQL editor

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

Analytical queries and associated transformations

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

Related tasks

[Creating visual queries using the Prompted Query editor](#)

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

[Creating visual queries in the Query Builder editor](#)

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

[Creating visual queries using the Draw Query wizard](#)

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

[Working with prompts in queries](#)

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

[Creating OLAP queries](#)

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

[Profiling queries](#)

You use profiling to analyze the complexity of the queries.

[Publishing objects to QMF Vision](#)

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

[Working with database tables](#)

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

Query interfaces

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

In the Visual Query Editor, you can create relational, analytical, and OLAP queries using different query interfaces that are tailored to different skill and knowledge levels. A technical user might select to use a familiar SQL interface, while a business user might opt for a graphical user interface.

The following table lists the different query editors supported in QMF and provides usage information about each editor.

| Type | Description | Characteristics of this type of editor |
|-----------------------|--|---|
| SQL Editor | Query is written in SQL. | <ul style="list-style-type: none">• Use to create visual queries.• SQL statements entered directly into editor.• User should be familiar with SQL. |
| Prompted Query Editor | Query is built by prompting the user for information about the tables, columns, rows, and conditions to use to create the query. | <ul style="list-style-type: none">• User guided through the query build process.• User does not need to be familiar with SQL.• Prompted Query editor constructs the SQL statements. |

| | | |
|----------------------|---|---|
| Query Builder Editor | Query is built visually by specifying table, join, column, sort, and row information. | <ul style="list-style-type: none"> • Use to create visual queries. • User diagrams the information used by the query. • User does not need to be familiar with SQL. • Query Builder editor constructs the SQL statements. |
|----------------------|---|---|

In the Visual Query Editor, you can switch between the editors by selecting the SQL, Prompted, or Build tabs in the query interface.

Tip: Actions taken in one tab automatically appear in the query views shown in the other tabs, allowing users to develop the query using a blend of the three query editors if necessary. However, certain changes made to the query text in the SQL editor (such as unions and subselects) might render the query unsuitable for the Prompted Query or Query Builder editors. In these cases, the query will remain editable in the SQL view only.

You can use the Draw Query wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

Experienced QMF users can use the command bar to access a query or database table. The command bar includes an area for the command itself, along with a means of selecting the server against which the command will run.

Related concepts

Query types

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

Creating visual queries using the SQL editor

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

Analytical queries and associated transformations

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

Related tasks

Creating visual queries using the Prompted Query editor

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

Creating visual queries in the Query Builder editor

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

Creating visual queries using the Draw Query wizard

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

Working with prompts in queries

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

Creating OLAP queries

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

Profiling queries

You use profiling to analyze the complexity of the queries.

Publishing objects to QMF Vision

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

Working with database tables

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

Creating visual queries using the SQL editor

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

Related concepts

Query types

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

Query interfaces

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

Analytical queries and associated transformations

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

Related tasks

Creating visual queries using the Prompted Query editor

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

Creating visual queries in the Query Builder editor

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

Creating visual queries using the Draw Query wizard

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

Working with prompts in queries

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

Creating OLAP queries

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

Profiling queries

You use profiling to analyze the complexity of the queries.

Publishing objects to QMF Vision

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

Working with database tables

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

Creating visual queries using SQL

You can create visual query objects in QMF.

About this task

This task describes how to create visual queries using SQL.

To create visual queries using SQL:

Procedure

1. Open the **Create New Visual Query** wizard by selecting **File > New > Other** and expanding the **QMF Objects** folder or by clicking the **New Visual Query** icon on the toolbar. If you open the wizard by clicking the **New Visual Query** toolbar icon, proceed from step 5.
2. Select **Visual Query** from the list of QMF Objects.
3. Specify the name of the visual query in the **Name** field and where the visual query will be saved in the **Data Source** field.
4. Click **Finish**.

The Visual Query editor opens.

5. Click the **SQL** tab to open the SQL input area in the editor.
6. Type the text of the SQL statement in the input text area.

You can type several distinct SQL statements for your query. You are responsible for ensuring that the SQL statements are syntactically correct.

Starting from QMF 12.2, you can use the XMLTABLE clause in your queries. Data retrieved by the XMLTABLE clause is available to work with on the **SQL**, **Design**, and **Display** tabs.

7. Click the **Run Query**  icon on the toolbar.

The visual query is run against the data source. Results are returned to the interface. The query results are displayed in the **Results** tab. You can switch between the multiple views by clicking the corresponding tabs.

If you typed two or more distinct SQL statements, the query returns a separate result set for each SQL statement. The list of result sets is available in the combo box on the toolbar of the **Design**, **Display**, and **Results** tabs.

8. Once you have obtained query results you have several options.
 - Switch result sets by using the **Result set** list at the top of the screen.
 - Format the columns and rows of the query results
 - Aggregate columns or rows
 - Add calculated columns to the query results
 - Define variables using a substitution window
 - Generate charts
 - Generate quick reports
 - Export the query results to numerous formats

Note: The **SQL visual query object** stays open until you close it. Upon closing, if you want to run the visual query object again you must save it to repository.

9. Click the **Save** icon to open the **Save object** window. You can save the visual query object to a file, repository, or to the QMF Catalog.

Using Content Assist for visual queries

If you are using QMF for Workstation, Content Assist helps to complete SQL statements by providing lists of options while the statement is being typed.

About this task

To use Content Assist while typing a SQL statement:

Procedure

1. Select **File > New > Other** and expand the **QMF Objects** folder. Select **Visual Query** from the list of QMF Objects.
2. Specify the name of the query in the **Name** field and where the query will be saved in the **Data Source** field. Click **Finish**.
3. Click the **SQL** tab.
4. At any time while typing the SQL statement, press **Ctrl+Spacebar**. A drop-down list opens underneath the cursor.
The contents of the drop-down list depend on the location of the cursor in the SQL statement and the contents of the table being called. For example, if `SELECT * FROM Q.` is displayed when Content Assist is selected, the drop-down list will display all tables in the data source with the owner name Q. If `SELECT * FROM Q.STAFF WHERE` is displayed, the drop-down list will display all columns of the Q.STAFF table.
5. If you select Content Assist while partially typing the name of a column, a function, or a table reference, and the partially typed keyword is not ambiguous, Content Assist will automatically complete the keyword.
For example, if you use Content Assist while typing `SEL`, the keyword `SELECT` will be completed automatically.
6. Select an option from the drop-down list. The selected text is automatically inserted into the SQL statement.

Using Parameter Hints for visual queries

If you are using QMF for Workstation, Parameter Hints provide a summary of a function's parameters as it is being typed in SQL statements.

About this task

To use Parameter Hints while typing parameterized functions into SQL text:

Procedure

1. Select **File > New > Other** and expand the **QMF Objects** folder. Select **Visual Query** from the list of QMF Objects.
2. Specify the name of the query in the **Name** field and where the query will be saved in the **Data Source** field. Click **Finish**.
3. Click the **SQL** tab.
4. Type the first part of a parameterized function, which includes the name of the function and the left parenthesis.
For example, `SUM(`.
5. Press **Ctrl+Shift+Spacebar**. A ToolTip opens that displays an example of how to complete the function.
For example, if `SUM(` is displayed, the ToolTip will display `SUM(<numeric-expression>)`.
6. Use the displayed hint as a reference when completing the parameterized function.



Creating visual queries using the Prompted Query editor

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

About this task

To create a new SQL query using the **Prompted Query** editor:

Procedure

1. Select **File > New > Other** and expand the **QMF Objects** folder.
 - a) Double-click **Visual Query**.
 - b) Specify the name of the query in the **Name** field and specify where the query will be saved in the **Data Source** field.
 - c) Click **Finish**.
 - d) Click the **Prompted** tab.
2. Use each of the following panes to build your query:
 - **Tables** where you specify the tables that will be used in the query.
 - **Join conditions** where you specify the join conditions for tables if you are using multiple tables in the query.
 - **Columns** where you specify the columns that will be included in the query results.
 - **Sort conditions** where you specify how the rows in the query results will be sorted.
 - **Row conditions** where you specify conditions that will limit the rows that are returned in the query results.
 - **Include duplicate rows** check box where you specify that duplicate rows will be included in the query results.
3. To add a distinct SQL statement to your query, click  in the toolbar.
4. To remove an unwanted SQL statement, click  in the toolbar.
5. When you are done building the query, click the **Run Query** icon.

The query is run against the currently connected to data source. Results are returned.

The query results are identified with a **Results** tab.

You can switch between the multiple views of a query by clicking the corresponding tabs.

Once you have obtained query results you have several options. You can:

- Format the query results' columns and rows.
- Aggregate columns or rows.
- Add calculated columns to the query results.
- Generate reports
- Export the query results to numerous formats.

The query object stays open until you close it. Upon closing, if you want to run the query object again you must save it.

6. Click the **Save** icon to open the **Save object** window.

Depending on the permissions granted to your user ID, you can save the query object to a file, a workspace, the QMF catalog, or the repository.

Related concepts

Query types

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

Query interfaces

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

Creating visual queries using the SQL editor

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

Analytical queries and associated transformations

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

Related tasks

Creating visual queries in the Query Builder editor

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

Creating visual queries using the Draw Query wizard

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

Working with prompts in queries

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

Creating OLAP queries

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

Profiling queries

You use profiling to analyze the complexity of the queries.

Publishing objects to QMF Vision

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

Working with database tables

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

Adding tables

You use the **Tables** window to add a table to a query when you are building the query using the **Prompted Query** editor. You can also use this dialog to change the tables that have been specified in the query.

About this task

To add a table to a query when you are building the query using the **Prompted Query**:

Procedure

1. Open the **Tables** window from the **Prompted Query** editor.
2. Identify the table that you want to include in the query. Type the table owner in the **Table owner** field.
3. Click the **Add Table** toolbar button in the **Tables** pane.
4. Type the table name in the **Table name** field.
5. If you do not know the table owner and name, you can search the data source to see what tables are available. To search for a table:
 - a) Specify the owner identifier of the tables that you want to include in the list in the **Table Owner** field. You must specify the owner in the correct format for identifiers. You can specify a matching pattern to match multiple names. To list all tables use the % sign.
 - b) You can further narrow the tables listed by using the name field. Specify an identifier for the table names that you want to include in the list in the **Table Name** field. You must specify the name in the correct format for identifiers. You can specify a matching pattern to match multiple names. To list all tables for a particular owner use the % sign.

- c) Click the **Add From List** button to search the data source for tables that match the criteria that you specified. The **Table List** window opens listing all the tables on the data source that met your search criteria.
 - d) Select a table to include in the query from the list of tables.
6. If you want to use the specified owner name for each successive added query, select the **Save Owner Filter** check box. Each time that you open the **Tables** dialog, box the saved owner name will automatically be displayed in the **Table owner** field.
 7. Click **Add**. The table is added to the query. The name of the table appears in the **Tables** pane of the **Prompted Query** editor.
If you add at least one more table to the query, the **Join Tables** dialog opens. In the dialog, you must specify the table joining option. For more information about table joining options, see the [Joining tables](#) topic.

Note: The **Join Tables** dialog appears only if the tables in your query are not joined in the data source ER Diagrams.

Joining tables

You can add more than one table. When you add an additional table to a query you must specify how the new table will be joined to an existing table.

About this task

When you add an additional table to a query using the **Tables** window, the **Join Conditions** dialogs open automatically.

There are two **Join Conditions** dialogs:

- The **Join Tables** window, where you specify what type of join will link the tables, such as an inner join or left, right or full outer join.
- The **Join Columns** window, where you specify the columns from each table that will be joined.

To join a table:

Procedure

1. Open the **Join Tables** window using the **Prompted Query** editor.
2. Select the table that is to be joined to the original table from the **Select a table to join into the query** list of tables.
3. Select the type of join that will link the two tables. Select the radio button for one of the following types of joins:
 - **Inner join** to include in the query results only those rows with matching values in both tables. An **Inner join** is implicit if you do not specify any other join operator. An **Inner join** compares every row of the left table with every row of the right table keeping only the rows that match. The resulting table might be missing rows from both of the joined tables.
 - **Left outer join** to include in the query results all the rows in the left table and only the rows from the right table that match with rows from the left table.
 - **Right outer join** to include in the query results all the rows in the right table and only the rows from the left table that match with rows from the right table.
 - **Full outer join** to include in the query results all rows from both tables.
4. Click **Continue** to complete the joining tables process.

What to do next

The step to joining tables is joining columns.

Joining columns

You must specify the columns that will be used to join the tables.

About this task

The **Join Columns** window opens automatically as the second phase to specifying join conditions for multiple tables when you are building a query using the **Prompted Query** editor. Rows in each of the tables that have equal values in these columns will be joined.

To join columns:

Procedure

1. Open the **Join Columns** window using the **Prompted Query** editor.
2. Select one column from the **Available columns to join** list.

The columns from all the tables that have been included in the query are listed.

You must select a column with the same data type (NUMERIC, DATE, TIME) as the column selected in the **Columns of (table name)** list.

3. Select one column from the **Columns of (table name)** list.

Only the columns from the right table (the table that you are joining) are listed.

You must select a column with the same data type (NUMERIC, DATE, TIME, CHARACTER) as the column selected in the **Available columns to join** list.

4. Click **Add**.

Rows that have equal values in the columns that are listed in the **Available columns to join** list and the columns that are listed in the **Columns of (table name)** list will be joined.

The **Join Columns** window closes.

What to do next

The next step is to specify what columns will be in the query results.

Specifying columns for query results

You can specify the columns that are included in the query results.

About this task

You use the **Columns** window to specify what columns are included in the query results in the **Prompted Query** or **Query Builder** editor. By default, all the columns from a table that is included in the query are included in the query results. If you have multiple tables included in the query, all the columns from each table are included.

To specify the columns that are included in the query results:

Procedure

1. Open the **Columns** window by clicking the **Add Column** button .

- Each table that was added to the query is listed in the **Table** field.

If there are two or more tables, each table is prefixed by a letter, such as Q . STAFF (A) and Q . INTERVIEW (B) .

- All the table columns are listed in the **Column** field.
- The type of data that is contained in the column is listed in the **Type** field.
- The label that is associated with the column is listed in the **Label** field.

Labels for columns are system column headers or column text. They are used when the query is displayed or when the results are printed.

- Any comments that are associated with a column are listed in the **Comments** field.
2. You can include a column in the query results in one of the following ways:
 - a) Select a column from the list of available columns by clicking the name of the column as it appears in the **Column** field.

You can filter the listed columns by typing a necessary column name or label, which is associated with the column, in the text field in the **Value** area.
 - b) Select all the columns of a table by selecting the name of the table in the **Table** field.
 - c) Select all the columns of a table by selecting the value <All columns> in the **Column** field.
 - d) Enter an expression in the **Or, enter an expression here** field.

The expression that you enter is used to evaluate a column and determine whether it is included in the query results.

Tip: Click the ellipsis (...) to open an expression builder. The expression builder offers a palette of common elements that are used to create SQL expressions, such as column names, constants, functions, and operators. When you click the expression builder buttons, templates for expression elements are inserted into the expression fields.

3. From the **Summary function** list, select how you want to summarize the column in the query results.

You can specify the following summary options:

- **(None)** - No summarization is applied.
 - **Average** - The average of all values in the column is included in summary field.
 - **Count** - A count of all the rows is included in the summary field.
 - **Maximum** - The maximum value that was in the column is included in the summary field.
 - **Minimum** - The minimum value that was in the column is included in the summary field.
 - **Sum** - The sum of all the values in the column is included in the summary field.
4. To specify a new name for the column in the query results, type the new name in the **New column name** field.

Unless you specify a new name, the name of the column in the column header of the query results is displayed exactly as it appears in the **Column** field.
 5. Click **Add**.

The column is included in the query results. Repeat these steps for each column that you want to include in the query results.
 6. If you work in the **Prompted Query** editor, click the **Change** button if you want to change to a column that was added to the query results.

The change to the column is made, and the **Columns** window closes.
 7. Click **Close** when you finish adding all the columns that you want to be included in the query results. The **Columns** window closes.

Specifying sort conditions for query results

You can specify the sort conditions that determine the order that the rows are returned in the query results.

About this task

Use the **Sort Conditions** dialog to specify the sort conditions that will apply to the query results. The **Sort Conditions** dialog is used when building a query using the **Prompted Query** editor. Rows can be sorted in ascending (A-Z) or descending (Z-A) order. If you sort query results rows by more than one column, the

first column is ordered first, then the second column is ordered within the sort order defined for the first column.

To specify the sort conditions for the query results:

Procedure

1. Open the **Sort Conditions** window by clicking the **Add Sort Condition** button in the **Sort conditions** pane of the **Prompted Query** editor.
2. Specify the columns that will be used to sort the query results in one of the following ways:
 - Select one or more columns from the **Columns in the result set** list. The **Columns in the result set** lists the columns that are included in the query results. You can select one or more columns if you are adding a sort condition. You can select only one column if you are changing a sort condition.
 - Specify a conditional expression in the **Or, enter a condition here** field. Click the ellipsis (...) to open an expression builder that will help you build the conditional expression. The expression builder offers a palette of common elements used to create SQL expressions, such as column names, constants, functions, and operators. When you click the expression builder buttons, templates for expression elements are inserted into the expression fields.
3. Specify the sort direction.

Click **Ascending** to specify that the query results will be sorted in ascending (lowest-to-highest) order.

Click **Descending** to specify that the query results will be sorted in descending (highest-to-lowest) order.

The first column is used for the primary sort; subsequent columns will sort within the first.
4. Click **Add** to add the sort condition to the query results.

If you are changing a sort condition, click **Change**.

Repeat this process for each column that you want included in the sort conditions.

When you have finished selecting the columns that will be included in the sort conditions, click **Close**. The **Sort Conditions** window closes.

Specifying row conditions for query results

You can specify row conditions to limit the rows that are returned in the query results. Without row conditions, all qualifying rows will be returned from the query.

About this task

You use the **Row Conditions** dialog to specify the row conditions for the query results. The **Row Conditions** dialog is used when building a query using the **Prompted Query** editor.

To specify the row conditions for query results:

Procedure

1. Open the **Row Conditions** window by clicking **Add Row Condition** in the **Row conditions** pane of the **Prompted Query** editor.
2. If you are specifying more than one condition that must be met for a row to be included in the query results, you must indicate how each condition will be connected using the **Connector** radio buttons.
 - If you select **And**, then the current condition that you are specifying and the previous condition (listed in the Row conditions pane of the **Prompted Query** editor) must be met in order for the row to be included.
 - If you select **Or**, then the current condition or the previous condition can be met in order for the row to be included.

When you connect multiple conditions using both **And** and **Or**, then the rules of the data source determine how the conditions are processed.

3. Select a column from the **Select a column from the list** to examine for inclusion in the query results.

The **Select a column from the list** lists all the columns that will be included in the query results.

You can filter the listed columns by typing a necessary row name in the text field in the **Left Side** area.

4. To specify a conditional expression that will determine the column, enter an expression in the **Or, enter an expression here** field.
5. Click the ellipsis (...) to open an expression builder that will help you build the conditional expression.

The expression builder offers a palette of common elements used to create SQL expressions, such as column names, constants, functions, and operators.

When you click the expression builder buttons, templates for expression elements are inserted into the expression fields.

6. Specify the operator for the row condition expression using the **Operator** field.

- a) To use the operator as is, select **Is**.
- b) To change the operator to its negative opposite, click **Is Not**.

For example, to specify "Is Not Equal To" click **Is Not** and select the **Equal To** operator in the list. You cannot use the **Starting with**, **Ending with**, or **Containing** operators if the left and right sides of the row condition expression have numeric data types.

Note: If IBM Db2 Spatial Extender is installed on the database, you are able to work with spatial data types: points, lines, and polygons. But you cannot use any of the spatial predicates unless the left and right sides have spatial data types.

7. Specify the condition for which to check the rows in the **Right Side** area.
 - a) You can use the expression builder button to help you build the conditions.
 - b) To add row conditions from a file of any format with the values that are separated by commas, click **Import Conditions From File**.
 - c) To specify row conditions from a set of query results, click **Import Conditions via Executing SQL Query**, and enter the SQL text of a query.
 - d) You can use the **Remove Condition** or **Remove All Conditions** buttons to remove one or all row conditions.

8. Click **Add**.

The row condition is added and displayed in the **Row conditions** pane of the **Prompted Query** editor.

Repeat this process for each condition that you want to add.

9. When you have finished specifying each condition, click **Close**.

The **Row Conditions** window closes and control returns to the **Prompted Query** editor.

Creating visual queries in the Query Builder editor

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

About this task

To create a visual query in the **Query Builder** editor, you must add tables. Then you can add joins to them, to specify particular table fields, add row and global conditions, to edit the SQL text of the created query, or add multiple SQL statements to the query:

Related concepts

[Query types](#)

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

Query interfaces

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

Creating visual queries using the SQL editor

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

Analytical queries and associated transformations

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

Related tasks

Creating visual queries using the Prompted Query editor

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

Creating visual queries using the Draw Query wizard

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

Working with prompts in queries

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

Creating OLAP queries

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

Profiling queries

You use profiling to analyze the complexity of the queries.

Publishing objects to QMF Vision

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

Working with database tables

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

Adding and joining tables in the Query Builder editor

To create a visual query in the **Query Builder** editor, you must add tables first.



About this task

To add tables in the **Query Builder** editor:

Procedure

1. From the main menu, select **File > New > Visual Query**. The **Create New Visual Query** wizard opens. Specify the name of the query in the **Name** field and the data source that you want to use for the query in the **Data Source** field. Click **Finish**.
2. Click the **Build** tab. The **Query Builder** editor opens. The **Build** tab is divided into three sections: **Tables**, **Fields**, and **SQL Editor**.
Tip: Expand and collapse the sections by clicking buttons next to the section names to make all functions of a particular section available.
3. To add a table to the query:

- a) Click the **Plus** button.
 - b) In the **Tables** window, specify the table owner and the table name or click **Add from List...** to select a table or several tables from a specified list.
 - c) Click **Add**.
- You can view the added tables in the **Tables** section. If the specified data source contains ER diagrams, you can view the tables from the ER diagrams in the **Suggested Tables** group. If there are no available ER diagrams, the **Suggested Tables** group will contain all the tables that are stored under the table owner node from which you added the tables.
4. To add a table from **Suggested Tables**, double-click it.

Note: To view a long list of items, use the scroll bar. If the number of items in the **Suggested Tables** list exceeds 500, the tables cannot be viewed.
 5. To view the available ER diagrams from the specified data source and add or remove fields, click the **ER Diagrams** button.
 6. To remove a table from the query, navigate to the table that you want to remove and click **Close**.
 7. To add a new SQL statement to the query, click  in the toolbar.
 8. To remove the SQL statement from the query, click  toolbar button.
 9. To sort columns in the table, navigate to the table the columns of which you want to sort and click the sorting button. You can also clear the **Show All Columns** check box to hide the columns that are not included in the query.
 10. If you add several tables to your query, the **Join Tables** dialog opens. In the dialog, specify the joining option for the tables in your query. For more information about table joining options, see the [Joining tables](#) topic.

Note: The **Join Tables** dialog appears only if the tables in your query are not joined in the data source ER Diagrams.
 11. To specify relationships among several tables and join tables, select a column from one table and drag it to a column of another table. You can set up multiple join conditions. If the **Tables** section is expanded, a connecting line is drawn from the column in one table to the column in the second one.
 12. Optional: To edit the type of join conditions, in the expanded **Tables** section, right-click the join line that appears between any two tables. Select **Change Join** from the menu to change the join conditions. The **Join Tables** window opens. You can select new join conditions. For detailed information about types of join conditions, see the step 2 in the [Editing join conditions for multiple tables](#) topic.
 13. To remove the join condition, right-click the join that you want to delete and select **Remove Join**.

Editing join conditions for tables

You can edit the join conditions to alter how two tables are linked.

About this task

To edit the join conditions that have been specified for two tables:

Procedure

1. Right-click the line that joins the table diagrams in the **Tables** section of the **Query Builder editor** window. Click **Change Join**. The **Join Tables** window opens.
2. Select the type of join that will link the two tables from one of the following choices:
 - **Inner join** to include in the query results only those rows with matching values in both tables.
 - **Left outer join** to include in the query results all the rows in the left table and only the rows from the right table that match with rows from the left table.

- **Right outer join** to include in the query results all the rows in the right table and only the rows from the left table that match with rows from the right table.
 - **Full outer join** to include in the query results all rows from both tables.
3. Click **Change** to save the new join conditions.

Specifying fields and conditions for a visual query

In the **Fields** section of the **Query Builder** editor, you can select the columns that you want to include in the query result set and specify the sort and row conditions for the query results.

About this task

To specify table fields in the **Query Builder** editor:

Procedure

1. Expand the **Fields** section on the **Build** tab.

Note: All columns of all the tables that are selected for the query are automatically included in the query results column table. The <All columns> entry displays when all table columns are included in the query.

- To display particular columns in the query results, you can select or clear the corresponding check boxes.
 - To add individual columns to the query results, drag a column from one of the tables in the **Tables** section to a particular row in the query results table in the **Fields** section.
2. To specify particular columns to a query, see the [“Specifying columns for query results”](#) on page 64 topic.
 3. For each column that you included in the column table, you can apply sort and row conditions in the corresponding table fields. Click one of the cells in the column to specify a value for one or more of the following columns:

| Column Name | Action |
|---------------------|--|
| Table | The column specifies the name of the table that includes the current column. |
| Column Name | The column specifies the name of the current column. Tip: Double-click the Column Name cell, and then click the down arrow. All column names from the current tables are listed. You can select another column name to replace the current column or remove it. |
| Column Type | The column displays the type of data that the current column contains. |
| Display Name | The column specifies the name that is used as the column header in the query results. |
| Aggregation | The column specifies the type of aggregation that is applied to the column. Double-click the Aggregation cell, and click the down arrow. |
| Sort Order | The column specifies whether this column is used to sort the rows in the query results. Double-click the Sort Order cell, click the down arrow, and select the sorting. |

| <i>Table 6. Columns and available actions (continued)</i> | |
|---|--|
| Column Name | Action |
| Sort Sequence | The column specifies the sort order sequence for the column. The column with the lowest number is sorted first. |
| Conditions | The column specifies a condition that must be met for a row of data to be included in the query results. The condition that you specify in this column applies to the column value. For example, if you had a column that is called Number and you entered a row condition of > 10, the only rows of data that are included in the query results are those rows that have a value greater than 10 in the Number column. Type the row conditions that you want to apply to each column. |

4. To specify whether to add duplicate rows to the query results, select the **Include duplicate rows** check box.

5. To view the index information of the tables added to the **Tables** section, click **Show Indexes**.

Index information of the of the tables added to the **Tables** section is displayed.

To hide the index information, you can click the **Hide Indexes** button.

Specifying columns for query results

You can specify the columns that are included in the query results.

About this task

You use the **Columns** window to specify what columns are included in the query results in the **Prompted Query** or **Query Builder** editor. By default, all the columns from a table that is included in the query are included in the query results. If you have multiple tables included in the query, all the columns from each table are included.

To specify the columns that are included in the query results:

Procedure

1. Open the **Columns** window by clicking the **Add Column** button .

- Each table that was added to the query is listed in the **Table** field.

If there are two or more tables, each table is prefixed by a letter, such as Q . STAFF (A) and Q . INTERVIEW (B) .

- All the table columns are listed in the **Column** field.
- The type of data that is contained in the column is listed in the **Type** field.
- The label that is associated with the column is listed in the **Label** field.

Labels for columns are system column headers or column text. They are used when the query is displayed or when the results are printed.

- Any comments that are associated with a column are listed in the **Comments** field.

2. You can include a column in the query results in one of the following ways:

- a) Select a column from the list of available columns by clicking the name of the column as it appears in the **Column** field.

You can filter the listed columns by typing a necessary column name or label, which is associated with the column, in the text field in the **Value** area.

- b) Select all the columns of a table by selecting the name of the table in the **Table** field.

c) Select all the columns of a table by selecting the value <All columns> in the **Column** field.

d) Enter an expression in the **Or, enter an expression here** field.

The expression that you enter is used to evaluate a column and determine whether it is included in the query results.

Tip: Click the ellipsis (...) to open an expression builder. The expression builder offers a palette of common elements that are used to create SQL expressions, such as column names, constants, functions, and operators. When you click the expression builder buttons, templates for expression elements are inserted into the expression fields.

3. From the **Summary function** list, select how you want to summarize the column in the query results.

You can specify the following summary options:

- **(None)** - No summarization is applied.
- **Average** - The average of all values in the column is included in summary field.
- **Count** - A count of all the rows is included in the summary field.
- **Maximum** - The maximum value that was in the column is included in the summary field.
- **Minimum** - The minimum value that was in the column is included in the summary field.
- **Sum** - The sum of all the values in the column is included in the summary field.

4. To specify a new name for the column in the query results, type the new name in the **New column name** field.

Unless you specify a new name, the name of the column in the column header of the query results is displayed exactly as it appears in the **Column** field.

5. Click **Add**.

The column is included in the query results. Repeat these steps for each column that you want to include in the query results.

6. If you work in the **Prompted Query** editor, click the **Change** button if you want to change to a column that was added to the query results.

The change to the column is made, and the **Columns** window closes.

7. Click **Close** when you finish adding all the columns that you want to be included in the query results. The **Columns** window closes.

Specifying global row conditions

In the **Query Builder** editor, you can specify global row conditions for a visual query.

About this task

The global row conditions affect more than a specific column in the query results. You can use this pane if you want to set a condition that includes the data column that is available in the file, but is not included in the query results.

To specify global row conditions:

Procedure

1. Click **Show Row Conditions** and open the Global Conditions pane.

2. Click **Insert Condition**.

The **Row Conditions** window opens.

3. If you specify more than one condition that must be met for a row to be included in the query results, you must indicate how each condition is connected in the **Connector** area.

- **And** connector means that the current condition and the previous condition must be met in order for the row to be included.

- **Or** connector means that either the current condition or the previous condition can be met in order for the row to be included.

Tip: When you connect multiple conditions with both **And** and **Or**, the rules of the data source determine how the conditions are processed.

4. In the **Left Side** area, select a column from the list that you want to use in the conditional expression for adding to the query results.

This list contains all the columns that are included in the query results. You can filter the listed columns by typing a necessary row name in the text field.

5. If you want to specify a conditional expression that determines the column, enter an expression in the **Or, enter an expression here** field.

- a) Click the ellipsis (...) to open an expression builder that can help you build the conditional expression.

The expression builder offers a palette of common elements that are used to create SQL expressions, such as column names, constants, functions, and operators.

- b) Click the expression builder buttons, templates for expression elements are inserted into the expression fields.

6. Select the operator for the global row condition expression in the **Operator** area.

- a) To use the operator as is, select **Is**.
- b) To change the operator to its negative opposite, click **Is Not**.

For example, to specify "Is Not Equal To" click **Is Not** and select the **Equal To** operator from the list.

Important: You cannot use the **Starting with**, **Ending with**, or **Containing** operators if the left and right sides of the row condition expression have numeric data types.

Important: If IBM Db2 Spatial Extender is installed on the database, you are able to work with spatial data types: points, lines, and polygons. But you cannot use any of the spatial predicates unless the left and right sides have spatial data types.

7. Specify the condition for which to check the rows in the **Right Side** area.

- a) You can use the expression builder button to help you build the conditions.
- b) To add row conditions from a file of any format with the values that are separated by commas, click **Import Conditions From File**.
- c) To specify row conditions from a set of query results, click **Import Conditions via Executing SQL Query**, and enter the SQL text of a query.
- d) You can use the **Remove Condition** or **Remove All Conditions** buttons to remove one or all row conditions.

8. Click **Add**.

The row condition is added and displayed in the Global Conditions pane of the **Query Builder** editor. Repeat this process for each condition that you want to add.

9. When you finish specifying each condition, click **Close**.

10. In the Global Conditions pane you can do the following actions:

- a) To change the order of the entered conditions, select a condition from the list and click **Move Condition Up** or **Move Condition Down**.
- b) To insert a condition before a particular condition, navigate to the required condition, click **Insert Condition**, and enter the text of the condition in a blank line. The condition is added to the list before the selected one.
- c) To edit a command, select it from the list, click **Edit Condition**, and change the text of the condition.
- d) To remove a condition, select it from the list, click **Remove Condition** or delete the text of the condition from the line.

Editing SQL text

The changes that you make in the **Tables** and **Fields** sections are automatically applied in the **SQL Editor** section to a visual query.

About this task

To edit the SQL text of a query in the **Query Builder** editor:

Procedure

1. Expand the **SQL Editor** section on the **Build** tab.
2. Edit the SQL text of a query.
3. Click the **Run Query** toolbar button. The query is run against the currently connected data source. The query results are displayed on the **Results** tab.

Creating visual queries using the Draw Query wizard

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

About this task

The **Draw Query** wizard builds the SQL statements for the query.

To create a query using the **Draw Query** wizard:

Procedure

1. Open the **Create New Query Using the Draw Query** wizard by clicking the **Draw Query** toolbar button.
You can also open the **Create New Query Using the Draw Query** wizard by selecting **File > New > Other > Query using Draw Query wizard**.
2. From the **Data Source** list select the data source where the tables that you want to use in the query are stored.
3. Click one of the **Query Type** radio buttons to select the type of query that you want to create. You can select one of the following:
 - **Select** to create a SQL SELECT statement that will retrieve rows of data from one or more tables.
 - **Update** to create a SQL UPDATE statement that will allow you to change data that is contained in a table.
 - **Insert** to create a SQL INSERT statement that will add new rows to a table. You can only update one table at a time using this statement.
4. In the **Query Tables** list specify the table or tables that will be included in the query.
 - a) To specify a table, type the owner of a table in the **Owner** field, the name of the table in the **Name** field, and the correlation ID of the table in the **Identifier** field.
 - b) Click **Add**.
The table is listed in the **Tables** list.
5. If you do not know the table name, you can search a list of tables that are available on the data source, by clicking **Add from list**.
The **Object List** window opens. Using the **Object List** window you can select one or more tables to include in the query if you have chosen the **Select** option as the query type.
6. To remove a table from the **Tables** list, select the table name and click the **Remove Table** icon. Use the **Move Table Up in List** or the **Move table Down in List** icons to move a table up and down in the **Tables** list.

7. Click **Finish** to create the query.

The **Create New Query Using Draw Query Wizard** closes. The SQL for the query is displayed in the query workstation window. You can edit the SQL in the available window.

Related concepts

Query types

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

Query interfaces

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

Creating visual queries using the SQL editor

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

Analytical queries and associated transformations

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

Related tasks

Creating visual queries using the Prompted Query editor

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

Creating visual queries in the Query Builder editor

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

Working with prompts in queries

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

Creating OLAP queries

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

Profiling queries

You use profiling to analyze the complexity of the queries.

Publishing objects to QMF Vision

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

Working with database tables

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

Listing data source tables

Use the **Objects List** dialog box to list the tables that are available on the data source.

About this task

To list the tables on a data source:

Procedure

1. To open the **Objects List** dialog, box click the **Add from list** button that appears in the **Draw Query** wizard.

2. Enter the owner of the table that you want to list in the **Owner** field and enter the name of the table that you want to list in the **Name** field.

You must specify the name in the correct format for identifiers. You can specify a matching pattern to match multiple names. Use the percent character (%) to match a string of any length containing any characters. For example, to list all tables with a name beginning with the letter A, enter A%. Use the underscore character (_) to match a single character. For example, to list all tables with an owner that has the letter A in the second position, enter _A%.

3. Click the **Refresh List** button to initiate a search for the tables on the database that match the search criteria that you specified in the **Owner** and **Name** fields.

Only those tables that you are authorized to use are included in the list.

The results are returned in the **Object List** list box. From this list select the table or tables that you want to use in your query.

4. Click **OK** to close the **Object List** dialog box.

Control returns to the **Draw Query** wizard. The table or tables that you selected are listed in the **Tables** list.

Working with prompts in queries

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

About this task

Use the following types of prompts depending on the object type:

| Object type | Available prompts |
|-------------------------------------|---|
| Visual query, dynamart | SQL prompts, simple prompts, and prompt hierarchies |
| Analytical query, compound dynamart | Shared prompts that consist of simple prompts or prompt hierarchies |

The following topics contain the detailed information about adding prompts:

Related concepts

[Query types](#)

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

[Query interfaces](#)

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

[Creating visual queries using the SQL editor](#)

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

[Analytical queries and associated transformations](#)

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

Related tasks

[Creating visual queries using the Prompted Query editor](#)

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

[Creating visual queries in the Query Builder editor](#)

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

Creating visual queries using the Draw Query wizard

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

Creating OLAP queries

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

Profiling queries

You use profiling to analyze the complexity of the queries.

Publishing objects to QMF Vision

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

Working with database tables

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

Working with SQL prompts

You use SQL prompts to limit the data retrieved from a visual query or a dynamart.

About this task

SQL prompts are part of SQL statements that you add when you write a condition to select only the data that is filtered by a specified value. When you run a query, you must specify the value for this prompt before you get the result set.

To add an SQL prompt to a visual query:

Procedure

1. Open the **SQL** tab.

Note: You can specify a prompt in the Conditions column on the **Build** tab of the Visual Query Editor.

2. Write a condition with a variable that is marked by an ampersand (&) character.
3. Type the name of the prompt after the ampersand.

Example

For example:

```
SELECT DISTINCT A.AREA, A.STATE_NAME
FROM DEMO.STATEDetails A
WHERE (A.AREA >&S)
```

Working with simple prompts

You use simple prompts to limit the data retrieved from a query. Simple prompts are closely related to a query and can be saved within it.

By determining simple prompt values you can filter query results and display only those that are based on the set values. For example, you run a query against company staff data and want to display data for a specific department only. You add a simple prompt to the query and every time you run it, you are asked to input the name of the department that you are interested in.

When you create a query with a substitutional parameter and use simple prompts with default values, the values are added automatically to the query and you are not prompted to enter them manually. Query results contain data only for the department name that is specified as the simple prompt default value.

To work with simple prompts, use either the **Manage prompts** window or the **Prompts** pane on the **Design** tab of the **Visual Query** editor.

Adding simple prompts

You can add simple prompts to a visual query, dynamart, analytical query, or compound dynamart to determine the data that is retrieved in this query so that only the specified value or ranges are used.

About this task

Use the **Simple Prompt** window to specify a prompt and add it to a query.

To add a simple prompt to a query:

Procedure

1. In the query editor, switch to the **Design** tab.
2. Use the **Prompts** pane on the **Design** tab to manage prompts.
Note: In visual queries or dynamarts, the table in the **Prompt Assignments** pane lists the columns of the query and prompts assigned to these columns. In analytical queries, the **Prompts** pane lists the shared prompts that can be used for this query. For more information about shared prompts, see [Working with shared prompts](#).
3. To create a simple prompt, click **New Prompt** on the **Prompts** pane toolbar.
4. Specify the name of the prompt in the **Prompt name** field of the **Simple Prompt** window.
5. In the **Display string** field, type the prompt text that is displayed when you run the query.
6. From the **Input type** list:
 - Select **Literal** for values that are enclosed within the double quotation marks if it is required by the database.
 - Select **Date** for values with a date part but no time part
 - Select **Time** for values with a time part but no date part.
 - Select **Timestamp** for values that contain both date and time parts.
 - Select **As is** for values that are passed to the database as they are entered, without any modification.
 - Select **Enumeration** for values that are represented as a set of predefined constants. See the example of use at the end of the topic.
7. If you selected *Time*, *Timestamp*, or *Date* option:
 - You can specify the format string in the **Format** field. You can either choose one of standard formats or specify your own string that corresponds to Date and Time Patterns.
 - You can set the value in the **Date and Time** window. To open the window, click **Choose default date or time**.
8. If you need a certain value to be used automatically when you run the query, select the **Has default value** check box and specify that value in the **Default value** field.
Note: If the **Has default value** check box is cleared, you are prompted to input the value when you run the query. If it is selected, the specified default value is used automatically and the prompt window is not displayed.
9. Assign the prompt to a query column by selecting one of available columns from the **Assigned column** list.
10. To edit a simple prompt, select a prompt that you want to edit from the list of available prompts on the **Prompts** pane and then click **Edit Prompt**.
11. To delete a simple prompt, select it from the list of available prompts in the **Prompts** pane and click **Delete Prompt**.
12. Click **OK** to close the **Simple Prompt** window.

Results

To view the prompt values that you specified when you last ran the query, select **Query > View Prompt Values**.

Example

Use the Enumeration variable type in functions that require an array as a parameter, for example, in the `in_op()` function with `getprompt()` used in the argument. The values that are entered in `getprompt()` are parsed and represented as an array for the database. Consider a query that is based on the Sample Data Source from the Samples repository:

```
SELECT SHOP_NAME, ADDRESS, CITY, STATE, ZIP, PHONE FROM DEMO.PIZZA
```

You can create a free style filter with the formula: `in_op(@[STATE], getprompt("state_arr"))`, and create the `state_arr` prompt of the enumeration type, then run the query and enter **'AK','AL'** in the **Prompt Variables** dialog, the result set shows data only for AK, AL states.

Related tasks

[Saving prompt values with a query](#)

The Prompt values that you enter when running a query can be saved to this query.

Saving prompt values with a query

The Prompt values that you enter when running a query can be saved to this query.

About this task

The **Save prompt values** function provides you with the ability to use the last saved value as a default in the **Prompt Variables** window next time you run this query.

When you use a dynamart or compound dynamart with prompts, the prompt values that you enter are always saved within this dynamart.

If you decide to save prompt values, be aware that any user can see the last prompt value in the **Prompt Variables** window and run the query with this value as default.

To save prompt values within a query:

Procedure

1. With an active query in the editor, select **Query > Manage Prompts**.
2. In the **Manage Prompts** window, select **Save prompt values** check box. If the check box is cleared, the prompt values are not saved and you are to enter the prompt value anytime you run the query. If the check box is selected, you can view the last value in the **Prompt Variables** window and use it as default.
Note: When you use prompts with dynamarts and compound dynamarts, the **Save prompt values** check box is always selected and cannot be cleared.
3. Click **OK** to close the window.
4. Run the query by clicking **Run Query** on the toolbar.
5. In the **Prompt Variables** window, input the value that you want the query to run with.
6. After the query is run, save it to a repository or file system.

Related tasks

[Adding simple prompts](#)

You can add simple prompts to a visual query, dynamart, analytical query, or compound dynamart to determine the data that is retrieved in this query so that only the specified value or ranges are used.

[Creating prompt hierarchies](#)

You can create a prompt hierarchy by using the **Create New Prompt Hierarchy** wizard.

Adding existing prompt hierarchies

You can add an existing prompt hierarchy that is stored in a repository or file to a visual query, analytical query, dynamart or compound dynamart.

Embedding prompt hierarchies in queries

You can create an embedded prompt hierarchy to store it with a visual query, dynamart, analytical query, or compound dynamart.

Working with prompt hierarchies

Prompt hierarchies allow users to select one or more values from a flat or hierarchical list of options.

Prompt hierarchies are distinct QMF objects that are stored independent of a given query, allowing them to be used by multiple query objects. Prompt hierarchies draw their option list from a query, for example “select region from sales territories”. In addition, a prompt hierarchy can present a hierarchy of values, with unlimited levels of detail. For example, “select region, country from sales territories”. In this case, the list of options would be broken down by ‘Americas’, ‘Europe’ and ‘APAC’, with USA, Canada and Mexico listed under ‘Americas’ and so on. Users can then select values at both levels of the prompt, perhaps selecting ‘APAC’ in its entirety and ‘Americas’ but not selecting Mexico.

Creating prompt hierarchies

You can create a prompt hierarchy by using the **Create New Prompt Hierarchy** wizard.

About this task

To create a prompt hierarchy:

Procedure

1. From the main menu, select **File > New > Other** to open the **Select a wizard** page of the **New** wizard.
2. Expand the **QMF Objects** folder, select **Prompt Hierarchy**, and click **Next** to open the **Create New Prompt Hierarchy** wizard.
3. Specify a name for the prompt hierarchy in the **Name** field and click **Finish** to close the **Create New Prompt Hierarchy** wizard and open the **Prompt Hierarchy** editor.
4. From the **Data Source** section of the **Prompt Hierarchy** editor, select the source of the data that you want to use for the prompt hierarchy. The data source for a prompt hierarchy can be a visual query, analytical query, dynamart, or compound dynamart.
 - Select **Linked**, if you want to store a link to a data source object in the prompt hierarchy that you create. Then go to step 5.
 - Select **Embedded**, if you want to save a data source within the prompt hierarchy that you create. Then go to step 6.
5. If you selected **Linked**, click the ellipsis (...) button that is next to the **Resource** field, and navigate to and select the required data source.

The path to the data source that you selected is displayed in the **Resource** field.
6. If you selected **Embedded**, select one of the following options:
 - Click **Import** to navigate to and select the required data source object.
 - Click **New** to create a data source query or dynamart for the prompt hierarchy.
 - Click **Edit** to modify the query or dynamart that is already specified as the embedded data source for the prompt hierarchy.
7. In the **Hierarchy levels** section, click **Add Level** to add levels to the prompt hierarchy.
8. From the **Hierarchy levels** list, select a level that you added and specify the following properties for it:
 - In the **Level name** field, specify the name that you want to use for the level.

- From the **Display value** list, select a column, which values are displayed when a user is prompted to enter the variable value.
 - From the **Return value** field, select a column that returns the variable value that corresponds to the display value.
For example, the "SALES" department has a department number of "10". The department name column is set as the display value, and the department number column is set as the return value. If the user enters SALES in the prompt, 10 is returned to the query.
 - From the **Order by** list, select a column that you want to use for ordering query results.
9. To change the position of the levels in the prompt hierarchy, select a level and click **Move Level Up** or **Move Level Down**.
 10. In the **Prompt hierarchy preview** section, click **Refresh** to preview the prompt hierarchy.
 11. When you finished adding levels to the prompt hierarchy, save it to a repository or file.

Results

To view the prompt values that you specified when you last ran the query, select **Query > View Prompt Values**.

Related tasks

[Adding existing prompt hierarchies](#)

You can add an existing prompt hierarchy that is stored in a repository or file to a visual query, analytical query, dynamart or compound dynamart.

[Embedding prompt hierarchies in queries](#)

You can create an embedded prompt hierarchy to store it with a visual query, dynamart, analytical query, or compound dynamart.

[Saving prompt values with a query](#)

The Prompt values that you enter when running a query can be saved to this query.


Adding existing prompt hierarchies

You can add an existing prompt hierarchy that is stored in a repository or file to a visual query, analytical query, dynamart or compound dynamart.

About this task


To add an existing prompt hierarchy:

Procedure

1. In the query editor, switch to the **Design** tab.
2. Click the arrow on the **New Prompt Hierarchy** () button and select **Add Prompt Hierarchy**.
3. In the **Prompt name** field, specify the name that you want to use for the prompt hierarchy.
4. In the **Display string** field, specify the text that you want to display when the query is run.
5. Click the ellipsis (...) button to navigate to and select the prompt hierarchy that you want to add.
The path to the prompt hierarchy that you selected is displayed in the **Resource** field.
6. Select the type of the resource.
 - Select **Embedded** to store the prompt hierarchy with the query or dynamart.
 - Select **Linked** to store the link to the prompt hierarchy in the query or dynamart.
7. Optional: If you add a prompt hierarchy to a visual query or dynamart, in the **Prompt Level** column, you can select the required level of the prompt hierarchy for each query column.
8. Click **OK** to close the **Prompt Hierarchy** dialog and return to the **Design** tab of the query editor.
The prompt hierarchy that you added is displayed in the **Prompts** pane.
9. Assign prompt hierarchy levels to selected columns:

- In a visual query or dynamart, assign the selected levels of the prompt hierarchy to the query columns:
 - a. In the **Prompt Assignments** pane, select a column.
 - b. Click the down arrow in the **Prompt** column to select the prompt hierarchy level for the column.
- In an analytical query or compound dynamart, add the prompt hierarchy to a group of shared prompts to assign it to the columns of the child queries:
 - a. In the **Transformations** pane, double-click **Filter** to open the **Dynamart Filter** wizard.
 - b. Select **Enable filtering**.
 - c. From the **Filter type** area, select **Prompt Hierarchy Filter**. Click **Next** to open the **Assign prompt hierarchy levels to columns** page.
 - d. Select a column and click the down arrow in the **Prompt** column to select the prompt hierarchy level for the column.
 - e. Click **Finish** to close the **Dynamart Filter** wizard.

Note: You can assign a prompt hierarchy to a calculated column that you added to the query.

10. To edit a prompt hierarchy, select it from the list of available prompts in the **Prompts** pane and click **Edit Prompt**.
11. To delete a prompt hierarchy, select it from the list of available prompts in the **Prompts** pane and click **Delete Prompt**.
12. To update data of the linked prompt hierarchy, select it in the **Prompts** pane and click **Refresh linked prompt hierarchy**.
13. To edit the level structure or data source object for the prompt hierarchy, select it in the **Prompts** pane and click **Open in prompt hierarchy editor** .

The prompt hierarchy is opened in the **Prompt Hierarchy** editor.

Related tasks

Creating prompt hierarchies

You can create a prompt hierarchy by using the **Create New Prompt Hierarchy** wizard.

Embedding prompt hierarchies in queries

You can create an embedded prompt hierarchy to store it with a visual query, dynamart, analytical query, or compound dynamart.

Saving prompt values with a query

The Prompt values that you enter when running a query can be saved to this query.

Embedding prompt hierarchies in queries

You can create an embedded prompt hierarchy to store it with a visual query, dynamart, analytical query, or compound dynamart.


Before you begin


Open a query to which you want to add an embedded prompt hierarchy on the **Design** tab of the query editor.

About this task

To create a prompt hierarchy and embed it in the query:

Procedure

1. On the **Design** tab, in the **Prompts** pane, click the down arrow and select **Create Embedded Prompt Hierarchy**  to open the **Create Embedded Prompt Hierarchy** wizard.

2. In the **Prompt name** field, specify the name that you want to use for the prompt hierarchy.
 3. In the **Display string** field, specify the text that you want to display when the query is run.
 4. Click **Finish** to close the **Create Embedded Prompt Hierarchy** wizard and open the **Prompt Hierarchy** editor.
 5. In the **Prompt Hierarchy** editor, create a prompt hierarchy and save it. For more information, see the [“Creating prompt hierarchies” on page 80](#).
 6. Return on the **Design** tab of the query editor.
The prompt hierarchy that you created is embedded in the query and displayed in the **Prompts** pane.
 7. Assign prompt hierarchy levels to selected columns:
 - In a visual query or dynamart, assign the selected levels of the prompt hierarchy to the query columns:
 - a. In the **Prompt Assignments** pane, select a column.
 - b. Click the down arrow in the **Prompt** column to select the prompt hierarchy level for the column.
 - In an analytical query or compound dynamart, assign the selected levels of the prompt hierarchy to the columns of the child queries:
 - a. In the **Transformations** pane, double-click **Filter** to open the **Dynamart Filter** wizard.
 - b. Select **Enable filtering**.
 - c. From the **Filter type** area, select **Prompt Hierarchy Filter**. Click **Next** to open the **Assign prompt hierarchy levels to columns** page.
 - d. Select a column and click the down arrow in the **Prompt** column to select the prompt hierarchy level for the column.
 - e. Click **Finish** to close the **Dynamart Filter** wizard.
 - Note:** You can assign a prompt hierarchy to a calculated column that you added to the query.
 8. To edit a prompt hierarchy, select it from the list of available prompts in the **Prompts** pane and click **Edit Prompt**.
 9. To delete a prompt hierarchy, select it from the list of available prompts in the **Prompts** pane and click **Delete Prompt**.
 10. To update data of the linked prompt hierarchy, select it from the list of available prompts in the **Prompts** pane and click **Refresh linked prompt hierarchy**.
 11. To edit the level structure or data source object for the prompt hierarchy, select it in the **Prompts** pane and click **Open in prompt hierarchy editor** ().
- The prompt hierarchy is opened in the **Prompt Hierarchy** editor.

Results

To view the prompt values that you specified when you last ran the query, select **Query > View Prompt Values**.

Related tasks

[Creating prompt hierarchies](#)

You can create a prompt hierarchy by using the **Create New Prompt Hierarchy** wizard.

[Adding existing prompt hierarchies](#)

You can add an existing prompt hierarchy that is stored in a repository or file to a visual query, analytical query, dynamart or compound dynamart.

[Saving prompt values with a query](#)

The Prompt values that you enter when running a query can be saved to this query.

Saving prompt values with a query

The Prompt values that you enter when running a query can be saved to this query.

About this task

The **Save prompt values** function provides you with the ability to use the last saved value as a default in the **Prompt Variables** window next time you run this query.

When you use a dynamart or compound dynamart with prompts, the prompt values that you enter are always saved within this dynamart.

If you decide to save prompt values, be aware that any user can see the last prompt value in the **Prompt Variables** window and run the query with this value as default.

To save prompt values within a query:

Procedure

1. With an active query in the editor, select **Query > Manage Prompts**.
2. In the **Manage Prompts** window, select **Save prompt values** check box. If the check box is cleared, the prompt values are not saved and you are to enter the prompt value anytime you run the query. If the check box is selected, you can view the last value in the **Prompt Variables** window and use it as default.

Note: When you use prompts with dynamarts and compound dynamarts, the **Save prompt values** check box is always selected and cannot be cleared.

3. Click **OK** to close the window.
4. Run the query by clicking **Run Query** on the toolbar.
5. In the **Prompt Variables** window, input the value that you want the query to run with.
6. After the query is run, save it to a repository or file system.

Related tasks

Adding simple prompts

You can add simple prompts to a visual query, dynamart, analytical query, or compound dynamart to determine the data that is retrieved in this query so that only the specified value or ranges are used.

Creating prompt hierarchies

You can create a prompt hierarchy by using the **Create New Prompt Hierarchy** wizard.

Adding existing prompt hierarchies

You can add an existing prompt hierarchy that is stored in a repository or file to a visual query, analytical query, dynamart or compound dynamart.

Embedding prompt hierarchies in queries

You can create an embedded prompt hierarchy to store it with a visual query, dynamart, analytical query, or compound dynamart.

Working with shared prompts

Use shared prompts to apply filtering to analytical queries and compound dynamarts.

About this task

In analytical queries, you use prompt groups that include shared child query prompts, newly added simple prompts, or prompt hierarchies. You can group prompts of the same type so that all the prompts within one prompt group receive the value passed to this group.

To manage shared prompts in analytical queries or compound dynamarts:

Procedure

1. With an active query in the query editor, either select **Query > Manage Prompts** or open the **Prompts** pane on the **Design** tab.
On the **Prompts** pane, you can see the child query prompts or newly added simple prompts or prompt hierarchies.
2. To add a simple prompt, perform the steps from [Adding simple prompts](#).
The newly added simple prompt is automatically added to a newly created prompt group that borrows its name.
3. To add an existing prompt hierarchy, perform the steps from [Adding existing prompt hierarchies](#).
The newly added prompt hierarchy is automatically added to a newly created prompt group that borrows its name.
4. To move a prompt from the default group to another group:
 - a) From the **Available prompts** list, select an item that you want to add to a group.
 - b) From the **Prompt Group** list, select a group to which you want to add the selected item.
 - c) Click **Add to Group**. The selected item is added to the prompt group.**Note:** You can only group prompts of the same type and prompt hierarchies with the same resource.
5. To remove a prompt from a group:
 - a) From the **Prompt Group** list, select an item that you want to remove.
 - b) Click **Detach Prompt**. The selected item is detached from the prompt group.
6. To rename a prompt group:
 - a) From the **Prompt Group** list, select the prompt that you want to rename.
 - b) Click **Rename Prompt Group**.
 - c) In the **Prompt Group** window, specify the name and string to display and click **OK**.
7. To change the order of the prompt groups in the **Prompt Group** list:
 - a) From the **Prompt Group** list, select the node of the prompt group or a prompt in the group.
 - b) Click **Move Prompt Group Up** or **Move Prompt Group Down**.
8. If you work in the Manage Prompts window, click **OK** to close the window.

Results

To view the prompt values that you specified when you last ran the query, select **Query > View Prompt Values**.

Saving prompt values with a query

The Prompt values that you enter when running a query can be saved to this query.

About this task

The **Save prompt values** function provides you with the ability to use the last saved value as a default in the **Prompt Variables** window next time you run this query.

When you use a dynamart or compound dynamart with prompts, the prompt values that you enter are always saved within this dynamart.

If you decide to save prompt values, be aware that any user can see the last prompt value in the **Prompt Variables** window and run the query with this value as default.

To save prompt values within a query:

Procedure

1. With an active query in the editor, select **Query > Manage Prompts**.

2. In the **Manage Prompts** window, select **Save prompt values** check box. If the check box is cleared, the prompt values are not saved and you are to enter the prompt value anytime you run the query. If the check box is selected, you can view the last value in the **Prompt Variables** window and use it as default.

Note: When you use prompts with dynamarts and compound dynamarts, the **Save prompt values** check box is always selected and cannot be cleared.

3. Click **OK** to close the window.
4. Run the query by clicking **Run Query** on the toolbar.
5. In the **Prompt Variables** window, input the value that you want the query to run with.
6. After the query is run, save it to a repository or file system.

Related tasks

Adding simple prompts

You can add simple prompts to a visual query, dynamart, analytical query, or compound dynamart to determine the data that is retrieved in this query so that only the specified value or ranges are used.

Creating prompt hierarchies

You can create a prompt hierarchy by using the **Create New Prompt Hierarchy** wizard.

Adding existing prompt hierarchies

You can add an existing prompt hierarchy that is stored in a repository or file to a visual query, analytical query, dynamart or compound dynamart.

Embedding prompt hierarchies in queries

You can create an embedded prompt hierarchy to store it with a visual query, dynamart, analytical query, or compound dynamart.

Creating OLAP queries

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

About this task

Using QMF for Workstation and QMF for WebSphere you can create an OLAP query for multidimensional data sources that is queried using MDX and accessed using XMLA.

Related concepts

Query types

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

Query interfaces

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

Creating visual queries using the SQL editor

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

Analytical queries and associated transformations

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

Related tasks

Creating visual queries using the Prompted Query editor

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

Creating visual queries in the Query Builder editor

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

Creating visual queries using the Draw Query wizard

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

Working with prompts in queries

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

Profiling queries

You use profiling to analyze the complexity of the queries.

Publishing objects to QMF Vision

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

Working with database tables

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

Retrieving cube data

OLAP queries are built based on a cube of multidimensional data. To create a new OLAP query you first must select and open a cube of data, then select the information from the cube that you want to view.

About this task

All the data cubes that are available for your use are listed in your workspace.

To retrieve cube data for an OLAP query:

Procedure

1. Open the **Create New OLAP Query** wizard.
2. On the **Select a data source** page, select the multidimensional data source where the cube that contains your data resides. Click **Next**.
3. The **Select a cube** page of the wizard opens. The cubes that are available on the data source are displayed in the **Cube models and cubes** list. Select a cube from the list. Click **Finish**.

The cube information is retrieved and displayed in layout format in the editor window. In the layout format, the **Cube Structure** tree lists the dimensions and measures that are available in the cube and the **Query Structure** tree lists the dimensions and measures that are selected for the OLAP query of the cube.

You modify the dimensions and measures that are listed in the **Query Structure** tree to build your OLAP query and obtain different views of the cube data. The structure of the OLAP query is also displayed in the **Outline** view.

4. When an OLAP query is open in the editor window, you can select one of the following tabs:
 - **Results** to open the **Results** page which displays the results of the current OLAP query. When you select the results tab, the OLAP query is rerun to update the results. You can use the **Outline** view and **Results** page to build OLAP queries dynamically.
 - **Filter** to open the **Filter** page where you can filter the dimensions and measures that will be retrieved from the cube.
 - **MDX** to open the **MDX** page which displays the MDX statements for the OLAP query. You can modify the MDX for the OLAP query.
 - **SQL** to open the **SQL** page which displays the SQL statements for the OLAP query. This is a read only view of the SQL that was created for the OLAP query. This page is only available if the

multidimensional data source uses the SQL-based API to view the SQL that has been created for the OLAP query.

- **Results XML** to open the XML page that displays the XML results for the OLAP query from the multidimensional data source.

This is a read-only view of the XML created for the OLAP query. This tab is available if the multidimensional data source works with XML for Analysis (XMLA).

- **Layout** to open the **Layout** page which displays all the dimensions and measures that are included in the cube and the dimensions and measures that are included in the current OLAP query. You use the **Layout** page to build an OLAP query offline.

Opening existing OLAP queries

You can open existing OLAP queries that are included in your workspace.

About this task

To open an existing OLAP query, expand and collapse folders in your workspace until you find the OLAP query that you want to open. Queries that have been created and saved as OLAP queries are identified with a unique icon containing a small cube in front of the query name. Open the OLAP query by double clicking the query. The query opens in an **OLAP Query editor** window.

Modifying OLAP queries to obtain different views of cube data

Once you have opened a cube of multidimensional data there are several ways to query the cube and expose the information that it contains.

Building the OLAP query offline

Building an OLAP query offline means that you will retrieve data from your cube data source only when you have finished building or making changes to your OLAP query.

About this task

You use the **Layout** page to build an OLAP query offline. You will initiate running the OLAP query when you select the **Results** tab.

To build an OLAP query offline:

Procedure

1. Double-click a cube in your workspace or open an existing OLAP query. The OLAP query opens in the editor window.
2. An OLAP query has four tabs that run across the bottom. Select the **Layout** tab. The **Layout** page opens. It includes the **Cube Structure** tree and the **Query structure** tree.
3. The **Cube Structure** tree lists the dimensions and measures that are contained in the cube.
4. The **Query Structure** tree lists the dimensions and measures that have been selected for inclusion in the OLAP query. The Query Structure tree consists of three elements:
 - **Top Dimensions:** The dimension values that are included in this node are displayed across the top of the query results with data summarized.
 - **Side Dimensions:** The dimension values that are included in this node are displayed along the side of the query results (starting in the first column) as a line of summary data.
 - **Measures** display in the query results as a column of data.
5. To select a measure for inclusion in the OLAP query, click the measure in the **Cube Structure** tree. With the mouse button pressed, drag the measure to the **Measures** node of the **Query Structure** tree.

Position the measure in the node in the same location as where you want the value to appear in the query results. Release the mouse button.

6. To select a dimension for inclusion in the OLAP query as a top or side dimension, click the dimension in the **Cube Structure** tree. With the mouse button pressed, drag the dimension to the **Top Dimensions** or **Side Dimensions** node of the **Query Structure** tree. Position the dimension in the node in the same order as you want its values to appear in the query results. Release the mouse button. Only dimensions can be used as top or side groups.
7. To remove a measure or dimension from the **Query Structure** tree, select it from the tree, drag it, and drop it back in the **Cube Structure** tree.
8. Click the **Results** tab to run the OLAP query. The OLAP query is run and the results are displayed on the **Results** page.

Related tasks

[Building an OLAP query online](#)

Building an OLAP query online means that you will retrieve data from your cube data source every time you make a change to an OLAP query.

[Filtering cube data](#)

Filtering enables you to limit the amount of data that is retrieved from a cube.

Building an OLAP query online

Building an OLAP query online means that you will retrieve data from your cube data source every time you make a change to an OLAP query.

About this task

You use the **Outline** view and the Results page to build an OLAP query online.

To dynamically build your OLAP query and see the results after each change:

Procedure

1. Double-click a cube in your workspace or open an existing OLAP query. The OLAP Query opens in the editor.
2. Open the **Outline** view. To open the **Outline** view, select **Window > Show views > Outline**.
3. A single structure tree is displayed in the **Outline** view or the **Layout** window. The dimensions and measures that are available in the cube are displayed under a node with the cube's name. The dimensions and measures that have been selected for the OLAP query are displayed under the **Layout** node.
4. To select a measure for inclusion in the OLAP query, click the measure in the cube's name node of the tree. With the mouse button pressed, drag the measure to the **Measures** node of the **Layout** node. Position the measure in the node in the same location as where you want the value to appear in the query results. Release the mouse button.
5. To select a dimension for inclusion in the OLAP query as a top or side dimension, click the dimension in the cube's name node of the tree. With the mouse button pressed, drag the dimension to the **Top Dimensions** or **Side Dimensions** node of the **Layout** node. Position the dimension in the node in the same order as you want its values to appear in the query results. Release the mouse button. Only dimensions can be used as top or side groups.
6. Each time you make a change, you will retrieve data from your cube data source and the query results are displayed in the **Results** page.

Related tasks

[Building the OLAP query offline](#)

Building an OLAP query offline means that you will retrieve data from your cube data source only when you have finished building or making changes to your OLAP query.

[Filtering cube data](#)

Filtering enables you to limit the amount of data that is retrieved from a cube.

Filtering cube data

Filtering enables you to limit the amount of data that is retrieved from a cube.

About this task

The **Filter** page lists the dimensions defined for the cube. Each dimension of a cube has one or more attributes. By selecting a check box you can choose to include or exclude the attribute's or entire dimension's data from being retrieved from a cube.

To filter the data that is retrieved from an OLAP cube:

Procedure

1. Double-click a cube in your workspace or open an existing OLAP query.
2. An OLAP query has six tabs that run across the bottom. Select the **Filter** tab. The **Filter** page opens. It includes the **Filter** tree and the **Filter Type** fields.
3. The **Filter** tree lists the dimensions of the cube, the hierarchy levels for each dimension, and the dimension values for each hierarchy level. Click the (+) to expand the dimension.
4. To select your filtering options, select the check boxes of the dimension values that you want included in the query. Each dimension, hierarchy level, and dimension value has a check box. You can expand and collapse the dimensions and hierarchies using the plus (+) and (-) boxes. You will select or clear the check boxes to include or exclude data beginning with the lowest level of the tree structure, the dimension values. If you select all values in a level, the higher level of the tree structure will be selected. If no values in a level are selected, the higher level of the tree structure will be cleared. If you select some of the values in a level, the higher level of the tree structure will be checked but with a grayed check box. You must select at least one value for each hierarchical level.
5. If the data cube resides on a multidimensional data source uses the SQL-based API, you can vary the organization of the tree depending on your selection of filter type.
 - Select **Regular Filter** to display a distinct list of dimension values by simple hierarchical level. All possible values are listed regardless of their hierarchy in the dimension. For example, if you have multiple quarters within multiple years, each dimension value would be organized as follows:

```
2001
2002
Q1
Q2
```

- Select **Hierarchical Filter** to organize the dimension values and display them under each of the parent hierarchy levels where they could possibly appear. This type of listing is useful when you have instances of a given value that can be included in multiple hierarchical levels. For example if you have multiple quarters within multiple years, each dimension value would be organized as follows:

```
2001
Q1
Q2
2002
Q1
Q2
```

6. Click the **Results** tab to run the OLAP query. Only the values that you checked will be included in the results.
7. To change the filtering, clear the check box next to each dimension item that you were filtering out from the OLAP query. Click the **Results** tab. The OLAP query is rerun and results are returned including only the values that you selected.

Related tasks

[Building the OLAP query offline](#)

Building an OLAP query offline means that you will retrieve data from your cube data source only when you have finished building or making changes to your OLAP query.

Building an OLAP query online

Building an OLAP query online means that you will retrieve data from your cube data source every time you make a change to an OLAP query.

Analytical queries and associated transformations

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

In analytical queries, you apply transformations to configure how result sets from multiple queries or tables are transformed into one query result set. There are several principle transformation types that you can incorporate into an analytical query structure:

- **Append** - to combine query results from two queries or tables into one query result set in such a way that the result set includes all values from both the queries or tables. You can specify how the columns will be paired up across the two result sets.
- **Union** - to combine query results from two queries or tables into one query result set in such a way that the result set includes unique values from the queries or tables. That is, to exclude duplicate values, use **Union**. You can specify how the columns will be paired up across the two result sets.
- **Except** - to combine query results from two queries or tables into one query result set in such a way that the result set includes all values from the first query or table and does not include any values from the second query or table. That is, to exclude all values from the second query or table, even if the values exist in the first query or table, use **Except**. You can specify how the columns will be paired up across the two result sets.
- **Intersect** - to combine query results from two queries or tables into one query result set in such a way that the result set includes common values that exist in both the queries or tables. That is, to only include the values that exist in both the queries or tables, use **Intersect**. You can specify how the columns will be paired up across the two result sets.
- **Join** - to join the result sets from two queries or tables into one unified result set by joining one or more columns together, much like a join operation in SQL.
- **Crosstab** - to perform a cross-tabulation of one or more columns, such as a sum, average, min, max, computed over two or more tabulation columns. For example, tabulating the sum of product sales across the region and month.
- **Normalize** - to reduce the number of columns in a table by returning data from eliminated columns in more rows. Normalizing query data facilitates query processing so that executions of the same query with different parameters can be easily compared with one another and aggregated together.
- **Conditional Grouping** - to create aggregated summaries of result set data based on conditional expressions.
- **Column Filter** - to add a calculated column before or after existing columns and enter filtering functions.

To create an analytical query, you perform the following procedures:

1. Building an analytical query structure. You select transformations that you want to apply in your analytical query.
2. Populating analytical query transformations. You add tables or queries that provide data for the transformations that you selected.
3. Configuring analytical query transformations. You configure the columns from the source data sets for the analytical query final result set.

Related concepts

Query types

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

Query interfaces

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

Creating visual queries using the SQL editor

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

Related tasks

Creating visual queries using the Prompted Query editor

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

Creating visual queries in the Query Builder editor

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

Creating visual queries using the Draw Query wizard

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

Working with prompts in queries

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

Creating OLAP queries

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

Profiling queries

You use profiling to analyze the complexity of the queries.

Publishing objects to QMF Vision

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

Working with database tables

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

Building analytical query structures

To create an analytical query, you begin by building a structure that arranges analytical query transformations in a tree of nodes.

About this task

By default, when you create an analytical query, the structure is simple and consists of one transformation, **Column Filter**. You can change it to a transformation of another type or expand the structure by adding transformations that can be both above and under the default transformation node in the tree. In this case, newly added transformations can encompass the current transformation or be child objects to it.

To build an analytical query structure:

Procedure

1. Select **File > New > Other**.
The **New** window opens.

2. Expand the **QMF Objects** folder, select **Analytical Query**, and click **Next**.

The **Create New Analytical Query** wizard opens.

3. Specify a name for the analytical query in the **Name** field.

4. Click **Finish**.

The **Create New Analytical Query** wizard closes and the new analytical query is displayed in the **Analytical query structure** with the **Column Filter** as a default transformation.

5. To change the current transformation to another:

- a. In the **Analytical query structure**, select the current transformation node.
- b. Click the **Change to...** icon.
- c. From the list, select the transformation type to which you want to change the current transformation.

The current transformation is changed to the selected transformation type and the **Item editor** pane displays the items of the new transformation.

6. If you want to include the current transformation into another transformation that will be placed above it in the structure tree:

- a. In the **Analytical query structure**, select the transformation node.
- b. On the toolbar in the **Item editor** pane, click the icon of the transformation that you want to add.

The new transformation is placed above the selected transformation in the analytical query structure and the **Item editor** pane displays the items of the new transformation.

7. If you want to add another transformation as a child object to a transformation in the current structure:

- a. In the **Analytical query structure**, select a node under the transformation.
- b. On the toolbar in the **Item editor** pane, click the icon of the transformation that you want to add.

The new transformation is placed under the selected transformation in the analytical query structure and the **Item editor** pane displays the items of the new transformation.

8. Make sure you have the required analytical query structure to proceed to populating it with data. Restructuring the analytical query leads to the loss of current transformation settings.

Related tasks

[Populating analytical query transformations](#)

After you finish building the structure of your analytical query, you populate it with data sets.

[Configuring analytical query transformations](#)

Configure transformations in your analytical queries to customize final result sets.

Populating analytical query transformations

After you finish building the structure of your analytical query, you populate it with data sets.

About this task

Analytical query structures consist of transformation nodes and data set nodes that must be populated with data from either queries or tables. Nodes that are not populated yet are displayed as **<none>** nodes. Regardless of what type of transformations you have in your analytical query structure, the rules for populating them with data are the same.

To populate an analytical query structure:

Procedure

1. In the **Analytical query structure**, select a transformation node that you want to populate with data. In the **Item editor** pane, you can view the items of the selected transformation.

2. From the **Repository Explorer** tree, navigate to a query or table that you want to add and drag it to a **<none>** node under the selected transformation node or directly into a corresponding column list in the transformation layout.

The columns of the selected data set are now displayed in the **Item editor** pane. The display of the columns depends on the type of the transformation that uses these columns.

Note: If you select a **<none>** node, you can drag a query or table anywhere in the blank area in the **Item editor** pane.

Note: You can drag a table or a query to a node that is already populated. This will replace the current data set with the new one.

Note: If you drag a table or a query to the transformation node, the transformation will be replaced with this table or query and the structure of the transformation will be lost.

3. To add a query that is currently open in a separate editor, right-click a **<none>** node in the analytical query structure, navigate to **Add from Opened...** and select one of the available queries.
4. To add a query that is saved in a data file:
 - a. Right-click a **<none>** node.
 - b. Select **Add Query**.
 - c. Select **Open from File** and click **Next**.
 - d. Browse for the query that you want to add and double-click it.
 - e. Click **Finish** to close the **Open from File** wizard.

The query from the data file is added to the analytical query structure.

5. To add a table that is saved in a data file:
 - a. Right-click a **<none>** node.
 - b. Select **Add Table**.
 - c. Select **Open from File** and click **Next**.
 - d. Browse for the table that you want to add and double-click it.
 - e. Click **Finish** to close the **Open from File** wizard.

The table from the data file is added to the analytical query structure.

6. To replace a query or table in the node with one of the queries that are opened in separate editors, select the node, click the **Change to...** icon. Navigate to **Change to Opened...** and select one of the available queries.
7. To replace a query or table in the node with a query or table from a data file:
 - a. Select the node.
 - b. Click the **Change to...** icon.
 - c. Navigate to **Change to Query** or **Change to Table**.
 - d. Click **Yes** in the **Change Item** window if you agree to continue.
 - e. Select **Open from File** and click **Next**.
 - f. Browse for the query or table that you want to add and double-click it.
 - g. Click **Finish** to close the **Open from File** wizard.

8. To rename a table or query, select it and click the **Rename** icon. Specify the new name for the query or table in the **Query name** field and click **OK**.
9. To detach a table or query, select it and click the **Detach** icon. A copy of the query or table opens in a separate editor. Any changes to this object have no effect on the analytical query that it was taken from.
10. To open and edit a query or table in a separate editor, select it and click the **Open in Separate Editor** icon. The query or table opens in a separate editor. Any changes to this object have a direct effect on the analytical query that it was taken from.

Note: You cannot view a query or table in the **Item editor** while it is opened in a separate editor.

11. You can create notes for any of the structure nodes in the **Note** section under the **Analytical query structure**:

- a) Select a structure node for which you want to enter a note text.
- b) Click anywhere in the **Note** section.
- c) Edit the existing note text or enter a new text.
- d) Save the changes to the note for the selected structure node by clicking **Save**.

When you highlight the structure node, you can view the new text in the **Note** section.

Note: Replacement of an element in the structure resets the note for the respective node to empty or changes the note text to the text of the newly selected analytical query.

Results

You have an analytical query structure of selected transformations and data sets. You can now work with each transformation individually to customize the result columns.

Related tasks

[Building analytical query structures](#)

To create an analytical query, you begin by building a structure that arranges analytical query transformations in a tree of nodes.

[Configuring analytical query transformations](#)

Configure transformations in your analytical queries to customize final result sets.

Configuring analytical query transformations

Configure transformations in your analytical queries to customize final result sets.

About this task

After you build an analytical query structure and populate it with data, you configure each of the transformations in the **Item editor** to define how the source columns are processed into the final result set. To configure analytical query transformations:

Procedure

1. In the **Analytical query structure**, select a transformation node that you want to configure.
In the **Item editor**, you can view the layout of columns for the selected transformation.
2. Configure the transformation by performing the procedure as described in the respective topic from the list below.
3. Repeat the steps for each of the transformations in your analytical query structure.

Related tasks

[Building analytical query structures](#)

To create an analytical query, you begin by building a structure that arranges analytical query transformations in a tree of nodes.

[Populating analytical query transformations](#)

After you finish building the structure of your analytical query, you populate it with data sets.

Working with append transformations

You use append transformations to add selected result set columns from different queries or tables to the final result set.

About this task

Append transformations allow you to combine result set columns from different queries or tables from either the same or different data sources to one result set. When you append two columns from two

different queries, all of the rows from both of the selected columns are combined in one column in the final query results. The column in the **First** list determines the name of the final query result set column, and the rows from the column will be displayed first.

To configure an append transformation in the analytical query structure:

Procedure

1. In the **Analytical query structure**, select an append transformation node.
In the **Item editor** pane, you can view the items of the append transformation. The columns from the source data set in the upper node are displayed in the **First** list. The columns from the source data set in the lower node are displayed in the **Second** list.
2. Optional: If the selected append transformation is not populated yet, add queries or tables into **<none>** nodes. For more information, see [Populating analytical query transformations](#).
3. Specify the query result columns that you want to add together.
 - a) Select a column from the **First** list.
 - b) Select a column from the **Second** list.
 - c) Click **Add Column** above the **Result columns** list.
The new append condition is added to the **Result columns** list.
 - d) Repeat this procedure for each append condition that you want to add.
4. If you are populating the append transformation with objects that contain similar names, you can try to automatically create the appropriate append conditions by clicking **Automatch**.
If there are any possible auto-matches, they are displayed in the **Result columns** list.
Note: Clicking **Automatch** will erase any append conditions that you previously set.
5. To include a column without matching it with any column from the other data set, select this column and select **<unmatched>** from the second column list. Click **Add Column** above the **Result columns** list.
6. Optional: To add another transformation to the current analytical query structure, follow the procedure described in [Building analytical query structures](#).

Results

Whether it is the only transformation in an analytical query or one of many nodes in a complex structure, an append transformation can be run on a database to accelerate the execution of this query. In the **Output** view, you can see that SQL UNION ALL is applied for such transformations. Generally, append transformations are run on databases if they contain queries with simple SQL statements using only the SELECT operator and retrieving data from one database. Db2 database queries can also apply SQL statements that involve join conditions, sort conditions, row conditions, calculated columns, simple filtering, groups, and categorizations.

Related tasks

[Working with union transformations](#)

You use union transformations to add unique result set columns from different queries or tables to the final result set.

[Working with except transformations](#)

You use except transformations to include values that are in one query or table and not in the other query or table to the final result set.

[Working with intersect transformations](#)

You use intersect transformations to include common values that exist in both the queries or tables to the final result set.

[Working with join transformations](#)

You use join transformations to join result set columns from different queries and tables.

[Working with crosstab transformations](#)

You use crosstab transformations to return aggregated data from a selected column across other columns.

Working with normalize transformations

You use normalize transformations to combine values from multiple columns in a single column by creating more rows.

Working with conditional grouping transformation

You use conditional grouping transformations to create aggregated summaries of result set data based on conditional expressions.

Working with column filter transformations

You use column filter transformations to add calculated columns and filters to the source query without modifying the source query itself. You can also choose to exclude columns from the result query.

Working with union transformations

You use union transformations to add unique result set columns from different queries or tables to the final result set.

About this task

Union transformations allow you to combine result set columns from different queries or tables from either the same or different data sources to one result set. When you perform union on two columns from two different queries, unique rows from both of the selected columns are combined in one column in the final query results. You can use **Collate Option** for case sensitive or case insensitive comparison of data. The **Collate** option gets activated only when the **Result Columns** entry is of the type string, var char, or text. The column in the **First** list determines the name of the final query result set column, and the rows from the column is displayed.

To configure a union transformation in the analytical query structure:

Procedure

1. In the **Analytical query structure**, select an union transformation node.
In the **Item editor** pane, you can view the items of the union transformation. The columns from the source data set in the upper node are displayed in the **First** list. The columns from the source data set in the lower node are displayed in the **Second** list.
2. Optional: If the selected union transformation is not populated yet, add queries or tables into **<none>** nodes. For more information, see [Populating analytical query transformations](#).
3. Specify the query result columns that you want to add together.
 - a) Select a column from the **First** list.
 - b) Select a column from the **Second** list.
 - c) Click **Add Column** above the **Result Columns** list.
The new union condition is added to the **Result Columns** list.
 - d) Repeat this procedure for each union condition that you want to add.

Note: Union of large object data types LOB, CLOB, and BLOB is not supported. An error message is displayed when you attempt to include a LOB, CLOB, or BLOB column in union transformations.
4. Optional: You can configure the result to be displayed in a case sensitive or case insensitive manner by clicking **Collate Option**. Specify **Collate Option** for each result column by selecting case sensitive or case insensitive option from the drop-down provided.

Note: **Collate Option** button is enabled when at least one of the result column is of the character data type.
5. If you are populating the union transformation with objects that contain similar names, you can try to automatically create the appropriate union conditions by clicking **Automatch**.
If there are any possible auto-matches, they are displayed in the **Result Columns** list.

Note: Clicking **Automatch** will erase any union conditions that you previously set.

6. To include a column without matching it with any column from the other data set, select this column and select **<unmatched>** from the second column list. Click **Add Column** above the **Result Columns** list.
7. Optional: To add another transformation to the current analytical query structure, follow the procedure described in [Building analytical query structures](#).

Results

Whether it is the only transformation in an analytical query or one of many nodes in a complex structure, an union transformation can be run on a database to accelerate the execution of this query. In the **Output** view, you can see that SQL UNION is applied for such transformations. Generally, union transformations are run on databases if they contain queries with simple SQL statements using only the SELECT operator and retrieving data from one database. Db2 database queries can also apply SQL statements that involve join conditions, sort conditions, row conditions, calculated columns, simple filtering, groups, and categorizations.

Note: When performing union transformations, you must ensure that the data in the columns, on which union is performed, are in the same format. That is:

- columns with textual data should be in the same case to be considered unique. For example, "IBM" and "ibm" are considered as distinct values.
- columns with numerical data should have the same precision and scale to be considered unique. For example, "12.010" and "0000012.01" are considered as distinct values.
- columns with date data should be in the same format to be considered unique. For example, "12/02/1980" and "12/02/80" are considered as distinct values.
- columns with currency data should be in the same format to be considered unique. For example, "\$100" and "\$100.00" are considered as distinct values.

To ensure that the union transformation is applied correctly, you can choose to use the database or QMF format functions to format the data.

Related tasks

[Working with append transformations](#)

You use append transformations to add selected result set columns from different queries or tables to the final result set.

[Working with except transformations](#)

You use except transformations to include values that are in one query or table and not in the other query or table to the final result set.

[Working with intersect transformations](#)

You use intersect transformations to include common values that exist in both the queries or tables to the final result set.

[Working with join transformations](#)

You use join transformations to join result set columns from different queries and tables.

[Working with crosstab transformations](#)

You use crosstab transformations to return aggregated data from a selected column across other columns.

[Working with normalize transformations](#)

You use normalize transformations to combine values from multiple columns in a single column by creating more rows.

[Working with conditional grouping transformation](#)

You use conditional grouping transformations to create aggregated summaries of result set data based on conditional expressions.

[Working with column filter transformations](#)

You use column filter transformations to add calculated columns and filters to the source query without modifying the source query itself. You can also choose to exclude columns from the result query.

Working with except transformations

You use except transformations to include values that are in one query or table and not in the other query or table to the final result set.

About this task

Except transformations allow you to combine result set columns from different queries or tables from either the same or different data sources to one result set. When you except two columns from two different queries, rows from only the first selected columns are included in one column in the final query results. You can use **Collate Option** for case sensitive or case insensitive comparison of data. The **Collate** option gets activated only when the **Result Columns** entry is of the type string, var char, or text. The column in the **First** list determines the name of the final query result set column, and the rows from the column is displayed.

To configure an except transformation in the analytical query structure:

Procedure

1. In the **Analytical query structure**, select an except transformation node.
In the **Item editor** pane, you can view the items of the except transformation. The columns from the source data set in the upper node are displayed in the **First** list. The columns from the source data set in the lower node are displayed in the **Second** list.
2. Optional: If the selected except transformation is not populated yet, add queries or tables into **<none>** nodes. For more information, see [Populating analytical query transformations](#).
3. Specify the query result columns that you want to add together.
 - a) Select a column from the **First** list.
 - b) Select a column from the **Second** list.
 - c) Click **Add Column** above the **Result Columns** list.
The new except condition is added to the **Result Columns** list.
 - d) Repeat this procedure for each except condition that you want to add.

Note: Except of large object data types LOB, CLOB, and BLOB is not supported. An error message is displayed when you attempt to include a LOB, CLOB, or BLOB column in except transformations.
4. Optional: You can configure the result to be displayed in a case sensitive or case insensitive manner by clicking **Collate Option**. Specify **Collate Option** for each result column by selecting case sensitive or case insensitive option from the drop-down provided.
Note: **Collate Option** button is enabled when at least one of the result column is of the character data type.
5. If you are populating the except transformation with objects that contain similar names, you can try to automatically create the appropriate except conditions by clicking **Automatch**.
If there are any possible auto-matches, they are displayed in the **Result Columns** list.
Note: Clicking **Automatch** will erase any except conditions that you previously set.
6. To include a column without matching it with any column from the other data set, select this column and select **<unmatched>** from the second column list. Click **Add Column** above the **Result Columns** list.
7. Optional: To add another transformation to the current analytical query structure, follow the procedure described in [Building analytical query structures](#).

Results

Whether it is the only transformation in an analytical query or one of many nodes in a complex structure, an except transformation can be run on a database to accelerate the execution of this query. In the **Output** view, you can see that SQL EXCEPT is applied for such transformations. Generally, except transformations are run on databases if they contain queries with simple SQL statements using only the SELECT operator and retrieving data from one database. Db2 database queries can also apply SQL statements that involve join conditions, sort conditions, row conditions, calculated columns, simple filtering, groups, and categorizations.

Note: When performing except transformations, you must ensure that the data in the columns, on which except is performed, are in the same format. That is:

- columns with textual data should be in the same case to be considered unique. For example, "IBM" and "ibm" are considered as distinct values.
- columns with numerical data should have the same precision and scale to be considered unique. For example, "12.010" and "0000012.01" are considered as distinct values.
- columns with date data should be in the same format to be considered unique. For example, "12/02/1980" and "12/02/80" are considered as distinct values.
- columns with currency data should be in the same format to be considered unique. For example, "\$100" and "\$100.00" are considered as distinct values.

To ensure that the except transformation is applied correctly, you can choose to use the database or QMF format functions to format the data.

Related tasks

Working with append transformations

You use append transformations to add selected result set columns from different queries or tables to the final result set.

Working with union transformations

You use union transformations to add unique result set columns from different queries or tables to the final result set.

Working with intersect transformations

You use intersect transformations to include common values that exist in both the queries or tables to the final result set.

Working with join transformations

You use join transformations to join result set columns from different queries and tables.

Working with crosstab transformations

You use crosstab transformations to return aggregated data from a selected column across other columns.

Working with normalize transformations

You use normalize transformations to combine values from multiple columns in a single column by creating more rows.

Working with conditional grouping transformation

You use conditional grouping transformations to create aggregated summaries of result set data based on conditional expressions.

Working with column filter transformations

You use column filter transformations to add calculated columns and filters to the source query without modifying the source query itself. You can also choose to exclude columns from the result query.

Working with intersect transformations

You use intersect transformations to include common values that exist in both the queries or tables to the final result set.

About this task

Intersect transformations allow you to combine result set columns from different queries or tables from either the same or different data sources to one result set. When you intersect two columns from two different queries, only common rows from both of the selected columns are combined in one column in the final query results. You can use **Collate Option** for case sensitive or case insensitive comparison of data. The **Collate** option gets activated only when the result column entry is of the type string, var char, or text. The column in the **First** list determines the name of the final query result set column, and the rows from the column is displayed.

To configure an intersect transformation in the analytical query structure:

Procedure

1. In the **Analytical query structure**, select an intersect transformation node.
In the **Item editor** pane, you can view the items of the intersect transformation. The columns from the source data set in the upper node are displayed in the **First** list. The columns from the source data set in the lower node are displayed in the **Second** list.
2. Optional: If the selected intersect transformation is not populated yet, add queries or tables into **<none>** nodes. For more information, see [Populating analytical query transformations](#).
3. Specify the query result columns that you want to add together.
 - a) Select a column from the **First** list.
 - b) Select a column from the **Second** list.
 - c) Click **Add Column** above the **Result Columns** list.
The new intersect condition is added to the **Result Columns** list.
 - d) Repeat this procedure for each intersect condition that you want to add.

Note: Intersect of large object data types LOB, CLOB, and BLOB is not supported. An error message is displayed when you attempt to include a LOB, CLOB, or BLOB column in intersect transformations.
4. Optional: You can configure the result to be displayed in a case sensitive or case insensitive manner by clicking **Collate Option**. Specify **Collate Option** for each result column by selecting case sensitive or case insensitive option from the drop-down provided.
Note: **Collate Option** button is enabled when at least one of the result column is of the character data type.
5. If you are populating the intersect transformation with objects that contain similar names, you can try to automatically create the appropriate intersect conditions by clicking **Automatch**.
If there are any possible auto-matches, they are displayed in the **Result Columns** list.
Note: Clicking **Automatch** will erase any intersect conditions that you previously set.
6. To include a column without matching it with any column from the other data set, select this column and select **<unmatched>** from the second column list. Click **Add Column** above the **Result Columns** list.
7. Optional: To add another transformation to the current analytical query structure, follow the procedure described in [Building analytical query structures](#).

Results

Whether it is the only transformation in an analytical query or one of many nodes in a complex structure, an intersect transformation can be run on a database to accelerate the execution of this query. In the **Output** view, you can see that SQL INTERSECT is applied for such transformations. Generally, intersect transformations are run on databases if they contain queries with simple SQL statements using only the SELECT operator and retrieving data from one database. Db2 database queries can also apply SQL statements that involve join conditions, sort conditions, row conditions, calculated columns, simple filtering, groups, and categorizations.

Note: When performing intersect transformations, you must ensure that the data in the columns, on which intersect is performed, are in the same format. That is:

- columns with textual data should be in the same case to be considered unique. For example, "IBM" and "ibm" are considered as distinct values.
- columns with numerical data should have the same precision and scale to be considered unique. For example, "12.010" and "0000012.01" are considered as distinct values.
- columns with date data should be in the same format to be considered unique. For example, "12/02/1980" and "12/02/80" are considered as distinct values.
- columns with currency data should be in the same format to be considered unique. For example, "\$100" and "\$100.00" are considered as distinct values.

To ensure that the intersect transformation is applied correctly, you can choose to use the database or QMF format functions to format the data.

Related tasks

Working with append transformations

You use append transformations to add selected result set columns from different queries or tables to the final result set.

Working with union transformations

You use union transformations to add unique result set columns from different queries or tables to the final result set.

Working with except transformations

You use except transformations to include values that are in one query or table and not in the other query or table to the final result set.

Working with join transformations

You use join transformations to join result set columns from different queries and tables.

Working with crosstab transformations

You use crosstab transformations to return aggregated data from a selected column across other columns.

Working with normalize transformations

You use normalize transformations to combine values from multiple columns in a single column by creating more rows.

Working with conditional grouping transformation

You use conditional grouping transformations to create aggregated summaries of result set data based on conditional expressions.

Working with column filter transformations

You use column filter transformations to add calculated columns and filters to the source query without modifying the source query itself. You can also choose to exclude columns from the result query.

Working with join transformations

You use join transformations to join result set columns from different queries and tables.

About this task

Join transformations allow you to join columns from different queries or tables from either the same or different data sources to one result set. You specify any number of join conditions, and specify inner, left, right, or full joins. You can use **Collate Option** for case sensitive or case insensitive comparison of data on the **Join Columns**. The **Collate** option gets activated only when the **Join Columns** entry is of the type string, var char, or text.

To configure a join transformation in the analytical query structure:

Procedure

1. In the **Analytical query structure**, select a join transformation node.
In the **Item editor** pane, you can view the items of the join transformation. The columns from the source data set in the upper node are displayed in the **Left Columns** list. The columns from the source data set in the lower node are displayed in the **Right Columns** list.
2. Optional: If the selected join transformation is not populated yet, add queries or tables into **<none>** nodes. For more information, see [Populating analytical query transformations](#).
3. On the toolbar in the **Item editor**, select the type of join.
 - **Inner join** to include in the query results only those rows with matching values in both tables.
 - **Left outer join** to include in the query results all the rows in the left table and only the rows from the right table that match with rows from the left table.
 - **Right outer join** to include in the query results all the rows in the right table and only the rows from the left table that match with rows from the right table.
 - **Full outer join** to include in the query results all rows from both tables.
4. Specify the query result columns that you want to join together.
 - a) Select a column from the **Left Columns** list.
 - b) Select a column from the **Right Columns** list.
 - c) From the **Join Columns** list, select the check box next to the join condition.
The new join condition is added to the **Join Columns** list and the **Condition(s) Added** check box is selected.
 - d) Repeat this procedure for each join condition that you want to add. For each new join condition, you can only use columns that were not previously selected.
 - e) To clear the **Join Columns** list, deselect the **Condition(s) Added** check box.
5. If you are populating a join transformation structure with objects that contain similar names, you can try to automatically create appropriate join conditions by clicking **Automatch**.
If there are any possible automatches, they are displayed in the **Join Columns** list.
Note: Clicking **Automatch** will erase any join conditions that you previously set.
6. Optional: You can configure the join condition to be applied in case sensitive or case insensitive manner by clicking on the **Collate Option** button. Specify **Collate Option** for each join column by selecting case sensitive or case insensitive option from the drop-down provided.
Note: **Collate Option** button is enabled when at least one of the join column is of the character data type.
7. To add a column to the **Result Columns** list, select the check box next to this column.

8. To move a column up or down the **Result Columns** list, select the column and click **Move Column Up** or **Move Column Down** buttons above the **Result Columns** list.
9. To add all of the **Left Columns** or **Right Columns** to the **Result Columns** list, select the **Include All Columns** check box above the respective column list.
10. To remove all of the **Left Columns** or **Right Columns** from the **Result Columns**, deselect the **Include All Columns** check box above the respective column list.
11. Optional: To add another transformation to the current analytical query structure, follow the procedure described in [Building analytical query structures](#).

Results

Whether it is the only transformation in an analytical query or one of many nodes in a complex structure, a join transformation can be run on a database to accelerate the execution of the entire analytical query. In the **Output** view, you can see that SQL Join is applied for such transformations. Generally, join transformations are run on databases if they contain queries with simple SQL statements using only the SELECT operator and retrieving data from one database. Db2 database queries can also apply SQL statements that involve join conditions, sort conditions, row conditions, calculated columns, simple filtering, groups, and categorizations. Join transformations with MS SQL database queries are always run locally.

Note: When performing join transformations, you must ensure that the data in the columns, on which join is performed, are in the same format. That is:

- columns with textual data should be in the same case to be considered unique. For example, "IBM" and "ibm" are considered as distinct values.
- columns with numerical data should have the same precision and scale to be considered unique. For example, "12.010" and "0000012.01" are considered as distinct values.
- columns with date data should be in the same format to be considered unique. For example, "12/02/1980" and "12/02/80" are considered as distinct values.
- columns with currency data should be in the same format to be considered unique. For example, "\$100" and "\$100.00" are considered as distinct values.

To ensure that the join transformation is applied correctly, you can choose to use the database or QMF format functions to format the data.

Related tasks

[Working with append transformations](#)

You use append transformations to add selected result set columns from different queries or tables to the final result set.

[Working with union transformations](#)

You use union transformations to add unique result set columns from different queries or tables to the final result set.

[Working with except transformations](#)

You use except transformations to include values that are in one query or table and not in the other query or table to the final result set.

[Working with intersect transformations](#)

You use intersect transformations to include common values that exist in both the queries or tables to the final result set.

[Working with crosstab transformations](#)

You use crosstab transformations to return aggregated data from a selected column across other columns.

[Working with normalize transformations](#)

You use normalize transformations to combine values from multiple columns in a single column by creating more rows.

[Working with conditional grouping transformation](#)

You use conditional grouping transformations to create aggregated summaries of result set data based on conditional expressions.

Working with column filter transformations

You use column filter transformations to add calculated columns and filters to the source query without modifying the source query itself. You can also choose to exclude columns from the result query.

Working with crosstab transformations

You use crosstab transformations to return aggregated data from a selected column across other columns.

About this task

Apply crosstab transformations to display the relations between the values in different columns by transforming one of the columns into a new row. When configuring a crosstab transformation, you define one or a few columns for the side group and one column for the top group. Then, you select an aggregated column to provide values for the newly formed fields in the intersections of top group and side group values.

For example, you have a table with data on production of a number of materials in different regions.

Table 8. Original result set data

| PRODUCT | QTY | REGIONID |
|-------------------|------------|-----------------|
| Plywood | 485 | 1 |
| Plywood | 712 | 2 |
| Plywood | 787 | 3 |
| Lumber | 959 | 1 |
| Lumber | 2372 | 2 |
| Lumber | 2360 | 3 |
| Siding | 1794 | 1 |
| Siding | 1940 | 2 |
| Siding | 739 | 3 |
| Industrial Panels | 1225 | 1 |
| Industrial Panels | 627 | 2 |
| Industrial Panels | 934 | 3 |

You want to cross-tabulate the QTY column data so that the aggregated values are related to the type of product and region number. You select and group the PRODUCT column as a side group column that will provide the rows and the REGIONID as a top group column to provide the columns. The aggregated values of the QTY column will be distributed across the fields that are formed by the intersections of the top group and the side group column values. As a result, you get the following table:

Table 9. Crosstab result set data

| PRODUCT | QTY(1) | QTY(2) | QTY(3) |
|-------------------|---------------|---------------|---------------|
| Industrial Panels | 1225 | 627 | 934 |
| Lumber | 959 | 2372 | 2360 |
| Plywood | 485 | 712 | 787 |

| Table 9. Crosstab result set data (continued) | | | |
|---|--------|--------|--------|
| PRODUCT | QTY(1) | QTY(2) | QTY(3) |
| Siding | 1794 | 1940 | 739 |

You can use **Collate Option** for case sensitive or case insensitive comparison of data. The **Collate** option gets activated when the following two conditions are satisfied:

- The selected column must be of the type string, var char, or text.
- Aggregation type should be **GROUP**.

To configure a crosstab transformation in the analytical query structure:

Procedure

1. In the **Analytical query structure**, select a crosstab transformation node.
In the **Item editor** pane, you can view the items of the crosstab transformation. The columns from the source data set are displayed in the **Group** and **Crosstab** sections.
2. Optional: If the selected crosstab transformation is not populated yet, add a query or table into the **<none>** node. For more information, see [Populating analytical query transformations](#).
3. In the **Group** section, select check boxes next to the columns that you want to use in the side group. In the example above, this is the PRODUCT column.

Note: Use the **Include All Columns** check box to select or deselect all available columns.

4. In the **Aggregation** column, click the value and select an aggregation type for the selected column from the list.
5. Optional: You can configure the grouping of records in case sensitive or case insensitive manner by clicking on the **Collate Option** button. Specify **Collate Option** for each group column by selecting case sensitive or case insensitive option from the drop-down provided.

Note: **Collate Option** button is enabled when **Grouping column** is of character data type and **Aggregation** is of type **Group**.

6. From the **Grouping column** list in the **Crosstab** section, select a column that will provide unique values to form columns in the top group. In the example above, this is the REGIONID column.
7. From the **Description column** list, select a column to provide values for the descriptions for the top group columns. In the example above, this is the REGIONID column.
8. Optional: You can configure the grouping of columns in case sensitive or case insensitive manner by clicking on the **Collate Option** button. Specify **Collate Option** for grouping column by selecting case sensitive or case insensitive option from the drop-down provided.

Note: **Collate Option** button is enabled when **Grouping column** is of character data type.

9. From the **Value column** column, select the check box next to the column that will provide the values for the crosstab cells. In the example above, this is the QTY column.
10. In the **Aggregation** column, click the value and select an aggregation type for the values in the crosstab cells.
11. Optional: To add another transformation to the current analytical query structure, follow the procedure described in [Building analytical query structures](#).

Related tasks

[Working with append transformations](#)

You use append transformations to add selected result set columns from different queries or tables to the final result set.

[Working with union transformations](#)

You use union transformations to add unique result set columns from different queries or tables to the final result set.

[Working with except transformations](#)

You use except transformations to include values that are in one query or table and not in the other query or table to the final result set.

Working with intersect transformations

You use intersect transformations to include common values that exist in both the queries or tables to the final result set.

Working with join transformations

You use join transformations to join result set columns from different queries and tables.

Working with normalize transformations

You use normalize transformations to combine values from multiple columns in a single column by creating more rows.

Working with conditional grouping transformation

You use conditional grouping transformations to create aggregated summaries of result set data based on conditional expressions.

Working with column filter transformations

You use column filter transformations to add calculated columns and filters to the source query without modifying the source query itself. You can also choose to exclude columns from the result query.

Working with normalize transformations

You use normalize transformations to combine values from multiple columns in a single column by creating more rows.

About this task

Each distinct original column value is placed within a new, single data column in the new result set, one row per column value. A second column is used to indicate which original column value a given row includes.

| Region | Revenue | Units Sold | Net Income |
|---------------|---------|------------|------------|
| North America | 250 | 452 | 87 |
| Asia Pacific | 290 | 538 | 92 |
| Europe | 320 | 675 | 120 |

Normalization allows us to transform the data into the following two forms:

| Region | KPI | KPI Value |
|---------------|------------|-----------|
| North America | Revenue | 250 |
| North America | Units Sold | 452 |
| North America | Net Income | 87 |
| Asia Pacific | Revenue | 290 |
| Asia Pacific | Units Sold | 538 |
| Asia Pacific | Net Income | 92 |
| Europe | Revenue | 320 |
| Europe | Units Sold | 675 |
| Europe | Net Income | 120 |

| Region | KPI | KPI Value | Units Sold |
|---------------|------------|-----------|------------|
| North America | Revenue | 250 | 452 |
| North America | Net Income | 87 | 452 |
| Asia Pacific | Revenue | 290 | 538 |
| Asia Pacific | Net Income | 92 | 538 |
| Europe | Revenue | 320 | 675 |
| Europe | Net Income | 120 | 675 |

In the first transformation, all three numeric columns have been normalized into a column pair. In the second sample, Revenue and Net Income have been normalized and Units Sold has been retained as an independent column. Note that the names of both the column that carries the value (KPI Value in this sample) and the column that indicates the original column (KPI) are arbitrary and can be defined by the user.

Normalized queries are often used to reverse an aggregation, rotating the pivot of the table and displaying individual rows for aggregated data. However, it is important to note that normalization can be applied to any result set data, not merely aggregated values.

Queries containing aggregated data can be normalized, in effect reversing the cross-tabulation and enabling the query results data to be displayed in a normalized format. For example, the Furniture Sales table is made up of the following columns Order Date, Category and Gross Profit, to display the gross profit of sales by the order date. The table is then crosstabbed to display a Gross Profit Column for each Category and an Order Date Column. The crosstabbed query results appear like the example shown below:

| Order Date | Gross Profit (CHAIRS) | Gross Profit (SOFAS) | Gross Profit (MISC) |
|------------------|-----------------------|----------------------|---------------------|
| 19 November 2008 | 160 | 452 | 87 |
| 23 November 2008 | 0 | 680 | 181 |
| 26 November 2008 | 120 | 642 | 0 |

Normalizing the query reverses the crosstab result set to produce query results that look like the original query results before the crosstab was applied. The normalized query results appear like the example shown below:

| Order Date | Category | Gross Profit |
|------------------|----------|--------------|
| 19 November 2008 | CHAIRS | 160 |
| 19 November 2008 | MISC | 87 |
| 19 November 2008 | SOFAS | 452 |
| 23 November 2008 | MISC | 181 |
| 23 November 2008 | SOFAS | 680 |
| 26 November 2008 | CHAIRS | 120 |
| 26 November 2008 | SOFAS | 642 |

To configure a normalize transformation:

Procedure

1. In the **Analytical query structure**, select a normalize transformation node.

In the **Item editor** pane, you can view the items of the normalize transformation. The original table columns are initially displayed in the **No Transform** section. The columns that need to be retained without any transformation, in the result set, should remain in **No Transform** section.

Transform: Columns which needs to be normalized must be moved to the **Transform** section.

Discard: Columns which needs not to be present in the result set must be moved to the **Discard** section.

2. Optional: If the node under normalize transformation is shown as none, add a query or table into the **<none>** node. For more information, see [Populating analytical query transformations](#).
3. To move the table columns from one section to another section, right click **Column Name > Move to Section**.

For example, if you want to move a column from **No Transform** section to **Transform** section, select the column and right-click > select **Move to Transform**.

Alternately, you can drag and drop the columns from one section to another section.

Note: You can select multiple columns by using the **Ctrl + Click**.

4. Click **Add** above the **Dimensions** list box.

The **Dimension Column** dialog opens. In **Dimension Column** dialog, specify the name and type of the column in the **Name** field and **Type** drop-down list respectively. This is the new column that will display the names of the normalized columns. In the example above, this is the **Category** column.

5. Optionally, you can check **Auto generate** check box to automatically create all the **Constants** for the dimension column.

In the example above, **Constants** are the different product type columns, such as CHAIRS, MISC, and SOFAS. These are labels for the normalized column names and as such can be anything. For example, if you wanted to make the query results more readable, you could specify lowercase names like chairs, misc, and sofas.

Note:

- **Auto generate** is enabled only for the character data type.
- **Automatch** will create dimension constants for all the original table columns.

6. Specify the data values for the dimension column under **Constants** section. If you have chosen to auto generate the data values of dimension columns, the auto generated data values will be displayed in **Constants** section.

7. Click **Add** above the **Values** list box.

The **Value Column** dialog opens. Specify the name and type of the column in the **Name** field and **Type** drop-down list respectively. This is the new column that will display the values of the normalized columns. In the example above, this is the **Gross Profit** column.

8. In the **Transform** section, to map each normalized column to its corresponding dimensions constant, click the **Value Identifiers** cell.

The **Dimension Mapping** dialog opens. In the **Dimension Mapping** dialog, for each **Dimension**, map the corresponding **Constants** for the particular normalized column.

9. Select the value from the **Value Column** drop-down for each normalized column.

10. Select the **Suppress Zeroes** or **Suppress Nulls** check boxes to exclude rows from the normalized result set that contains zero or no value.

11. Optional: If you want to generate dimensions, constants, value columns, and fill the dimension mapping of normalized columns automatically, click **Automatch** in the upper right corner of the **Item editor** and click **Yes** in the **Automatic filling the structure of normalize** dialog.

Note: **Automatch** will override all your existing query settings.

12. Optional: To add another transformation to the current analytical query structure, follow the procedure described in [Building analytical query structures](#).

Related tasks

Working with append transformations

You use append transformations to add selected result set columns from different queries or tables to the final result set.

Working with union transformations

You use union transformations to add unique result set columns from different queries or tables to the final result set.

Working with except transformations

You use except transformations to include values that are in one query or table and not in the other query or table to the final result set.

Working with intersect transformations

You use intersect transformations to include common values that exist in both the queries or tables to the final result set.

Working with join transformations

You use join transformations to join result set columns from different queries and tables.

Working with crosstab transformations

You use crosstab transformations to return aggregated data from a selected column across other columns.

Working with conditional grouping transformation

You use conditional grouping transformations to create aggregated summaries of result set data based on conditional expressions.

Working with column filter transformations

You use column filter transformations to add calculated columns and filters to the source query without modifying the source query itself. You can also choose to exclude columns from the result query.

Working with conditional grouping transformation

You use conditional grouping transformations to create aggregated summaries of result set data based on conditional expressions.

About this task

With conditional grouping transformations, you can create filters that control what values from a given value column are included in the aggregated summary. For example, you have a table that contains staff data for your company. You create a query that displays the sum of the salaries of all of the employees in a given department across each job type. Now, you want to only display the sum of the salaries of employees that have been working for five or more years. You can use a conditional grouping to create a filter that only includes in the aggregation salary values from rows with a value of 5 or more in the YEARS column.

| Department | Name | Years | Salary |
|------------|---------|-------|--------|
| SALES | Smith | 12 | 60 |
| SALES | Edwards | 7 | 40 |
| SUPPORT | Jones | 6 | 40 |
| R&D | Adams | 2 | 25 |
| R&D | Baker | 7 | 50 |

Using a conditional grouping, the result set can be transformed into the following:

| Department | Total Salary | Matched Employees | Total Employees |
|------------|--------------|-------------------|-----------------|
| SALES | 100 | 2 | 2 |
| SUPPORT | 40 | 1 | 1 |
| R&D | 50 | 1 | 2 |

This conditional grouping was done by carrying out the following steps:

1. Adding **Department** and **Name** as grouped columns. **Department** was added as a **GROUP** and **Name** was added as a **COUNT**. The **Name** column allows us to include a column showing total number of employees in a department, providing a contrast with the number used to perform the salary summation.
2. A filter was defined to select employees with 5 or more years service.
3. Two conditional columns were added, both driven by the above filter. **Total Salary** performs a sum of the **Salary** column and **Matched Employees** performs a count of the **Name** column.
4. Finally, the **Name** column in the result set was edited to display **Total Employees** via the **Design** tab (right-click the column and select **Change Column Heading** from the context menu).

To configure a conditional grouping transformation:

Procedure

1. In the **Analytical query structure**, select a conditional grouping transformation node.
In the **Item editor** pane, you can view the items of the conditional grouping transformation. The columns from the source data set are displayed in the **Group** section.
2. Optional: If the selected conditional grouping transformation is not populated yet, add a query or table into the **<none>** node. For more information, see [Populating analytical query transformations](#).
3. In the **Group** section, select check boxes next to the columns that will provide rows for grouping the aggregated results.
Note: Use the **Include All Columns** check box to select or deselect all available columns.
4. In the **Aggregation** column, click the value and select an aggregation type for the selected column from the list.
5. Optional: You can configure the grouping of records in case sensitive or case insensitive manner by clicking on the **Collate Option** button. Specify **Collate Option** for each group column by selecting case sensitive or case insensitive option from the drop-down provided.
Note: **Collate Option** button is enabled when **Grouping column** is of character data type and **Aggregation** is of type **Group**.
6. In the **Conditional Grouping** section, click **Add Filter**.
The **Filter Wizard** opens.
7. Specify a name for the filter in the **Filter name** field.
It is a good idea to give the filter a name that reflects its function, for example, **Older Group** for the filter that will display employees that have worked more than five years.
8. Select a filter type and click **Next**.
The next page of the wizard depends on your previous selection.
For more information about building filters, see:
 - [“Building a compound condition filter” on page 188](#)
 - [“Building a free style condition filter” on page 189](#)
 - [“Building a prompt hierarchy filter” on page 190](#)
9. Build your filter using the table of options or the expression designer and click **Finish**.

The **Filter Wizard** closes and the filter that you specified is displayed in the **Filters** table.

10. Above the **Columns** table, click **Add Column**.
11. From the list in the **Source Column** column, select the column that you want to filter.
This column will provide the values that will be filtered and aggregated across the group columns.
12. Select the filter that you will use with this column from the **Filter** list.
This list is populated with all of the filters that you built in the **Filters** table.
13. In the **Aggregation** column, click the value and select an aggregation type for the values in the selected column.
14. Specify a name for the column of aggregated values in the **Column name** field.
15. Once you have finished adding all of the filtered value columns that you want to display, click the **Run Query** toolbar button.
The query is run and the results are displayed in the **Results** editor. You will notice that there is a column for every group and value column that you selected. The value column headings display the column names that you specified. The contents of the value columns represent all of the values that are included in the conditional filters that you set and are aggregated according to the methods that you selected.
16. Optional: To add another transformation to the current analytical query structure, follow the procedure described in [Building analytical query structures](#).

Results

Related tasks

[Working with append transformations](#)

You use append transformations to add selected result set columns from different queries or tables to the final result set.

[Working with union transformations](#)

You use union transformations to add unique result set columns from different queries or tables to the final result set.

[Working with except transformations](#)

You use except transformations to include values that are in one query or table and not in the other query or table to the final result set.

[Working with intersect transformations](#)

You use intersect transformations to include common values that exist in both the queries or tables to the final result set.

[Working with join transformations](#)

You use join transformations to join result set columns from different queries and tables.

[Working with crosstab transformations](#)

You use crosstab transformations to return aggregated data from a selected column across other columns.

[Working with normalize transformations](#)

You use normalize transformations to combine values from multiple columns in a single column by creating more rows.

[Working with column filter transformations](#)

You use column filter transformations to add calculated columns and filters to the source query without modifying the source query itself. You can also choose to exclude columns from the result query.

Working with column filter transformations

You use column filter transformations to add calculated columns and filters to the source query without modifying the source query itself. You can also choose to exclude columns from the result query.

About this task

For example, you have a query that displays staff data for your company and that this query uses a calculated column to enumerate the rows. When you apply a filter to this query, some rows are excluded from the result set. You can use a column filter query to add another calculated column that displays the row numbers of the result query and not the source.

The following table shows a query with calculated column which returns row numbers:

| Deptname | Deptnumb | Division | Location | Manager | Row |
|----------------|----------|-----------|------------|---------|-----|
| HEAD OFFICE | 10 | CORPORATE | NEW YORK | 160 | 0 |
| NEW ENGLAND | 15 | EASTER | BOSTON | 50 | 1 |
| MID ATLANTIC | 20 | EASTER | WASHINGTON | 10 | 2 |
| SOUTH ATLANTIC | 38 | EASTER | ATLANTA | 30 | 3 |
| GREAT LAKE | 43 | MIDWEST | CHICAGO | 100 | 4 |
| PLAINS | 51 | MIDWEST | DALLAS | 140 | 5 |

If you apply a filter to this query, some rows will be dropped from the result set. But the calculated column will return the same row numbers as in the source query, as shown in the following table:

| Deptname | Deptnumb | Division | Location | Manager | Row |
|-------------|----------|-----------|----------|---------|-----|
| HEAD OFFICE | 10 | CORPORATE | NEW YORK | 160 | 0 |
| GREAT LAKE | 43 | MIDWEST | CHICAGO | 100 | 4 |
| PLAINS | 51 | MIDWEST | DALLAS | 140 | 5 |

You can use column filter query to add another calculated column that will show row numbers of the result query, as shown in the following table:

| Deptname | Deptnumb | Division | Location | Manager | Row | Row2 |
|-------------|----------|-----------|----------|---------|-----|------|
| HEAD OFFICE | 10 | CORPORATE | NEW YORK | 160 | 0 | 0 |
| GREAT LAKE | 43 | MIDWEST | CHICAGO | 100 | 4 | 1 |
| PLAINS | 51 | MIDWEST | DALLAS | 140 | 5 | 2 |

To configure a column filter transformation:

Procedure

1. In the **Analytical query structure**, select a column filter transformation node.
In the **Item editor** pane, you can view the items of the column filter transformation. The columns from the source data set are displayed in the **Result Columns** list.
2. Optional: If the selected column filter transformation is not populated yet, add a query or table into the **<none>** node. For more information, see [Populating analytical query transformations](#).

3. In the **Result Columns** list, use the **Include All Columns** check box to select or deselect all available columns.
4. To apply sorting conditions to the columns that you included to the **Result Columns** list, specify values in the **Sort Order** and **Key Sequence** columns.
 - In the **Sort Order** column, double-click the value and specify whether the column is used for sorting data in the query result set. Select the **Ascending**, **Descending**, or **None** sort order from the list for the appropriate column.
 - In the **Key Sequence** column, double-click the value and specify the sequence in which the sort order conditions are applied to the columns. Select the sequence number from the list for the columns with sort order conditions (the column that has the key sequence number '1' is sorted first and so on).
5. Click the **Design** tab. On the **Design** page you can manage prompts, calculated columns, and filters.
6. In the **Prompts** pane, you can add and edit prompts.
7. Use the **Calculated Column** window to add calculated columns. Click in the **Dynamart** pane, and then select **Add Calculated Column**.
8. Use the arrow buttons in the **Dynamart** pane to change the order of query columns. This column order is applied only when the query is used as a data source in the other application parts like dashboards or analytical queries.
9. Use the **Dynamart Filter** wizard to create conditional expressions that control the contents of query results. In the **Transformations** pane right-click the **Filter** node and select **Edit Filter**.

Note: When you use the **Transformations** pane controls the filtering is applied in the application, it does not change SQL text executed in the database.

For more information about adding calculated columns and building filters, see:

- [Adding calculated columns to the query results](#)
- [Filtering query results](#)

10. Optional: To add another transformation to the current analytical query structure, follow the procedure described in [Building analytical query structures](#).

Related tasks

Working with append transformations

You use append transformations to add selected result set columns from different queries or tables to the final result set.

Working with union transformations

You use union transformations to add unique result set columns from different queries or tables to the final result set.

Working with except transformations

You use except transformations to include values that are in one query or table and not in the other query or table to the final result set.

Working with intersect transformations

You use intersect transformations to include common values that exist in both the queries or tables to the final result set.

Working with join transformations

You use join transformations to join result set columns from different queries and tables.

Working with crosstab transformations

You use crosstab transformations to return aggregated data from a selected column across other columns.

Working with normalize transformations

You use normalize transformations to combine values from multiple columns in a single column by creating more rows.

Working with conditional grouping transformation

You use conditional grouping transformations to create aggregated summaries of result set data based on conditional expressions.

Profiling queries

You use profiling to analyze the complexity of the queries.

Before you begin

You must have a query to analyze open.



About this task

You can profile the following query types containing only SELECT statements:

- visual queries
- analytical queries

To profile a query:

Procedure

1. Open the required query in the Visual Query Editor.
2. Click the **Profile Query**  toolbar button.
The **Query Profiler** view opens. It displays the time spent on running the query and fetching its results if there are any.
3. Click the **SQL Text**  toolbar button in the **Query Profiler** view to see the SQL text of the last profiled query.
The SQL text is displayed in the **Query Profiler** window.

Related concepts

[Query types](#)

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

[Query interfaces](#)

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

[Creating visual queries using the SQL editor](#)

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

[Analytical queries and associated transformations](#)

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

Related tasks

[Creating visual queries using the Prompted Query editor](#)

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

[Creating visual queries in the Query Builder editor](#)

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

[Creating visual queries using the Draw Query wizard](#)

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

[Working with prompts in queries](#)

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

Creating OLAP queries

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

Publishing objects to QMF Vision

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

Working with database tables

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

Publishing objects to QMF Vision

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

About this task

The list of objects that you can publish to the QMF Vision Objects folder includes:

- Visual queries
- Analytical queries
- Prompted queries
- Dynamarts
- Compound dynamarts
- Tables

Note: You cannot import queries with prompts and substitution variables.

By publishing objects to QMF Vision, you save them to the QMF Vision Objects folder as links with data source information. This operation enables you to access the published objects from QMF Vision. While you perform the procedure, you must specify user information for objects that use data sources with login and password requirements to ensure the proper functioning of the published objects in QMF Vision. For more information about working with published objects in QMF Vision, see the *Getting Started with QMF Vision* manual.

To publish a query or table to QMF Vision:

Procedure

1. In the **Repository Explorer** or **Workspaces** view, right-click an object that you want to publish. To select several objects, use Ctrl or Shift keys.
2. Select **Publish to QMF Vision**.
3. Specify a destination folder in one of the following ways:
 - Select the default **QMF Vision Objects** folder.
 - Create a subfolder in **QMF Vision Objects**:
 - a. Click **New Folder**.
 - b. Specify the name of the folder.
 - c. To specify a different parent folder, you can choose a subfolder in **QMF Vision Objects** by clicking **Browse**.
 - d. Specify a comment that you can view in the properties of the subfolder.
 - e. Click **Finish**.

- Select one of the subfolders in **QMF Vision Objects**.
4. If you publish a single object, you can specify a new name.
 5. For objects with data sources that require login and password information, click **Next**. In the table, specify user credentials for each data source.
 6. Click **Finish**.

Related concepts

Query types

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

Query interfaces

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

Creating visual queries using the SQL editor

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

Analytical queries and associated transformations

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

Related tasks

Creating visual queries using the Prompted Query editor

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

Creating visual queries in the Query Builder editor

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

Creating visual queries using the Draw Query wizard

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

Working with prompts in queries

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

Creating OLAP queries

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

Profiling queries

You use profiling to analyze the complexity of the queries.

Working with database tables

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

Working with database tables

You can open any database table that is accessible to you in your workspace using a default editor, the **Table Editor**, the **Table Viewer** or the **Visual Query Editor**.

About this task

To open a database table:

Procedure

1. Right-click the table in any view and select **Open With**.
2. Click one of the following items:
 - **Table Viewer** to display the contents of the table in the **Query Results** view. When a table is opened with **Table Viewer**, a query is created, run and the results are returned.
 - **Visual Query Editor** to display the table in the **Query Builder** editor. When the table is opened with the **Visual Query Editor**, a query is created and displayed in the **Query Builder** editor. The query is not run and results are not returned. Users can modify the query using the **Query Builder** editor.
 - **Table Editor** to display the table in edit mode.

The default view for the Table Editor is the **Results** view, as indicated by the **Results** tab in the editing area.

Note: Permission to use the Table Editor is set by the administrator. Click **View > Resource Limits** and select the **Options** tab to verify whether you have permission to use the Table Editor. If you have permission to edit tables, the **Enable table editing** property is set to Yes.

You can use the **Table Editor** to search for, add, edit, and delete the data that is stored in the tables without writing SQL statements. You can also save the Table Editor query objects (SQL query along with the sorting and filtering options) for future use. You can open the saved Table Editor query object to resume working with the same SQL query that includes the previously used row conditions and sorting options. You can also save different types of row conditions and sorting options as multiple Table Editor query objects. This enables you to conveniently reuse the different views in future.

Using the Results menu item to work with database tables

The **Results** menu item is activated when viewing a table in the **Results** view of the Table Editor.

The **Results** menu provides access to most of the functionality in the Table Editor.

To access and use **Result** menu options:

- a. With the table opened in the **Table Editor**, select the **Results** tab.
- b. Click **Results** from the menu list to display available options.

The following table lists the **Results** menu options and describes the purpose of each option.

| Option | Purpose |
|--------------|--|
| Retrieve All | Select this option to retrieve all rows from the table. |
| Find | Select this option to search the table for specific data. Enter the search criteria in the Find dialog and click Find next . |
| Insert Row | Select this option to insert a table row. |
| Delete Row | Select this option to remove a table row. The row that you select for deletion is marked with an asterisk and is removed if you commit the change to the table. |
| Edit Row | Select this option to edit row data. |
| Commit | Select this option to manually commit the changes you have made to the table |
| Roll back | Select this option to undo changes that you have made to the table |

| Option | Purpose |
|------------------|--|
| Immediate Commit | Select this option to enable the immediate commit function, which results in changes to the table being committed immediately. |

Editing content directly in the table

You can edit table data directly in the **Table Editor** in the following ways:

- Placing your cursor in the cell containing the information you want to edit, right-clicking and selecting **Edit** from the pop-up menu.

In the **Cell value** window, enter your changes and click **OK**. Click **OK** again to confirm that the updates will be committed to the database.

Click **Cancel** if you want to roll back the updates.

- Placing your cursor in the cell and pressing **Enter** or by double-clicking in the table cell.

Changing the table structure

You change the structure of a table (for example, the number and data type of columns) directly in the **Table Editor** in the following ways:

- Deleting a row or multiple rows:
 - a. Click the row number of the row that you want to delete. To select multiple rows, press the **Shift** key while clicking the table rows.
 - b. Right-click and select **Delete Row** from the pop-up menu.
 - c. If you are sure that you want to delete the row from the table, click **Yes**. Click **No** if you decide not to delete the row from the table.

- Editing a table row:

- a. Click the row number of the row that you want to edit.
- b. Right-click and select **Edit Row** from the pop-up menu.

The row and its content are displayed in an **Edit Row** window. The **Edit Row** window title contains the row number from the results grid indicating the current row that is being edited.

- c. Make changes to data in the **Value** of the **Edit Row** window.

Note: The cell value to be edited might have Auto-Generated or Calculated values. These values are computed by the system and hence are disabled.

- d. You can choose to click the **Prev Row** and **Next Row** buttons on the **Edit Row** window to update the previous and next row contents of the table.

If the **Immediate Commit** option is selected on the **Results** menu, a **Confirm Updates** dialog box is displayed when you navigate to the previous or next rows of the table. On the **Confirm Updates** dialog box, click **OK** to commit the changes that you made to the current row, or click **Cancel** to ignore the changes that you made to the current row.

If the **Immediate Commit** option is cleared on **Results** the menu, all the changes that you make to the table are stored locally until you choose to commit the changes to the database using the **Commit** option on the **Results** menu. To disregard the locally stored changes, you can use the **Roll back** option on the **Results** menu.

Note: The **Results** tab is not updated or refreshed when navigating the rows of a table using the **Prev Row** and **Next Row** buttons. The updates that you make to the contents of a table are reflected in the **Results** tab only after you close the **Edit Row** window.

- e. After completing the desired updates to the table, click **OK** on the **Edit Row** window.
- f. If the **Immediate Commit** option is selected on the **Results** menu, a **Confirm Updates** dialog box is displayed. On the **Confirm Updates** dialog box, click **OK** to commit the changes that you

made to the current row, or click **Cancel** to ignore the changes that you made to the current row.

- Inserting a row:

Select **Insert Row** from the **Results** menu.

The **Insert Row** dialog box opens. The **Insert Row** dialog box might have cells marked as **<Auto-Generated>** or **<Calculated>**. These cells are disabled and you will not be able to specify the values for these cells, as these values are computed by the system.

- An **<Auto-Generated>** column generates a unique number when a record is inserted in the table.
- A **<Calculated>** column derives its value using an expression, rather than being specified through an insert or update operation.

If the option **Immediate Commit** on the menu is selected, the changes that you make to the table are applied to the database *as you make them* and click **OK**.

If the option **Immediate Commit** is cleared then the changes that you make are stored locally and are applied all at once when if you select the Commit option, or they are disregarded all at once if select the Roll back option.

Otherwise, when you close the editor object you will get a message to Commit or Rollback all pending changes.

The inserted row will be the last row of the result set after the commit is done.

Note: If you select a row before you click **Insert Row** then the **Insert** dialog will be pre-populated with the content of the row so that you can use it as a template for a new row.

To specify how to display the nonnumeric table data, place your cursor in the cell containing text string for which you want to set the display format. Right-click, select **Edit** from the pop-up menu and then click **Advanced**. From the **Format mode** list, select **Hexadecimal** to display the data in hexadecimal format or **Binary** to display the data in binary format. The **Zoom** window displays samples of the data in the format that you have chosen.

You can use the Prompted Query view of the Table Editor to set the sort and row conditions in the table. For information about setting sort and row conditions, see [Specifying sort conditions for tables](#) and [Specifying row conditions for tables](#).

- **Visual Query Editor** to display the table in the **Query Builder** editor. When the table is opened with the **Visual Query Editor**, a query is created and displayed in the **Query Builder** editor. The query is not run and results are not returned. Users can modify the query using the **Query Builder** editor.
- **Default Editor** to display the table using the editor that was last used.

Related concepts

[Query types](#)

QMF provides multiple ways to query data. What you can do with queries and the query results can vary depending on the type of query you run.

[Query interfaces](#)

QMF provides robust querying capabilities. You can create new queries using the query building editors or import existing queries. Imported queries can be edited as needed.

[Creating visual queries using the SQL editor](#)

You can create queries by typing SQL statements in the **SQL Query** editor. The **SQL Query** editor allows you to build visual queries.

[Analytical queries and associated transformations](#)

You use analytical queries to combine multiple data sets from the same or differing data sources into one result set.

Related tasks

[Creating visual queries using the Prompted Query editor](#)

You can build simple to complex queries using the **Prompted Query** editor. The **Prompted Query** editor prompts you for the information it needs to build the SQL statement.

Creating visual queries in the Query Builder editor

You can use the **Query Builder** editor to create visual queries. You supply table, join, column, sort, and row information and the editor constructs the SQL statements.

Creating visual queries using the Draw Query wizard

Use the **Draw Query** wizard to create a query object automatically based on the query type and tables that you specify. This menu option is the same as the DRAW command in QMF for CICS/TSO.

Working with prompts in queries

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

Creating OLAP queries

OLAP queries enable users to dynamically access various aspects of multidimensional data, including dimensions, hierarchies, members, titles, values, and instances, as well as rows, columns, and data points.

Profiling queries

You use profiling to analyze the complexity of the queries.

Publishing objects to QMF Vision

You can publish queries and tables to the QMF Vision Objects folder to work with them in QMF Vision.

Specifying sort conditions for tables

You can specify the sort conditions that determine the order that the rows are displayed in tables.

Related tasks

Specifying row conditions for tables

If you are using QMF for Workstation, you can specify row conditions to limit the rows that are displayed in the table. Without row conditions, all existing rows will be displayed. You can specify row conditions to limit the rows that are displayed in the table. Without row conditions, all existing rows will be displayed.

Editing database tables using Db2 special registers

You can change data in database tables using Db2 special registers. A special register is a storage area that is defined for an application process by Db2 and is used to store information for the reference. A reference to a special register is a reference to a value provided by the current server. The list of available special registers depends on the version of the database.

Copying tables

Use the **Repository Explorer** view to copy tables inside the currently open data source.

Using Prompted tab

About this task

Use the **Sort conditions** pane of the **Prompted** menu to set sort conditions for tables. The **Prompted** menu of the **Table** editor is similar to the **Prompted Query** editor, except that only the **Sort conditions** and **Row conditions** panes are available.

Procedure

1. Right-click a table in the **Workspaces** view and select **Open With > Table Editor**.
The table opens in the editor window.
2. Click the **Prompted** tab.
The **Prompted** menu opens with only the **Sort conditions** and **Row conditions** panes available.
3. Click the **Add Sort Condition** button in the **Sort conditions** toolbar.
The **Sort Conditions** window opens.

4. Specify the columns that will be used to sort the query results in one of the following ways:
 - Select one or more columns from the **Columns in the result set** list. The **Columns in the result set** lists the columns that are included in the table. You can select one or more columns if you are adding a sort condition. You can select only one column if you are changing a sort condition.
 - Specify a conditional expression in the **Or, enter a condition here** field. Click the ellipsis (...) to open an expression builder that will help you build the conditional expression. The expression builder offers a palette of common elements used to create SQL expressions, such as column names, constants, functions, and operators. When you click the expression builder buttons, templates for expression elements are inserted into the expression fields.
5. Specify the sort direction. Select **Ascending** to specify that the query results will be sorted in ascending (lowest-to-highest) order. Select **Descending** to specify that the query results will be sorted in descending (highest-to-lowest) order. The first column is used for the primary sort; subsequent columns will sort within the first.
6. Click **Add** to add the sort condition to the table. If you are changing a sort condition, click **Change**. Repeat this process for each column that you want included in the sort conditions. When you have finished selecting the columns that will be included in the sort conditions, click **Close**. The **Sort Conditions** window closes.

Using SQL tab

Procedure

1. Right-click a table in the **Workspaces** view and select **Open With > Table Editor**.
The table opens in the editor window.
2. Click the **SQL** tab.
The **SQL** tab opens displaying the SQL query that fetches the table displayed in the **Results** tab.
3. Specify the sort conditions:
 - Use the ORDER BY clause to specify the columns that will be used to sort the query results.
 - Use the ASC or DESC keywords to specify ascending or descending sort direction.

For example:

```
SELECT [Columns] FROM [Table_Name]
ORDER BY [Column 1] ASC|DESC, [Column 2] ASC|DESC
```

The first column is used for the primary sort; subsequent columns will sort within the first.

Note: Consider the following when specifying an SQL query in the **SQL** tab:

- Use an SQL query to fetch the contents of a single table only. You will not be able to use the Table Editor to update the resultant table if you query more than one table in your SQL queries. Use separate table editors to query and update tables individually.
 - Ensure that the SQL query that you specify fetches all the columns of the table. If all the columns of a table are not included in the result set, you may not be able to effectively use the Table Editor to update the table. You may encounter errors when you try to insert new rows into such tables. Therefore, it is recommended to avoid introducing new columns or omitting existing columns in the SQL query. Because joins, aggregate functions, and calculated columns may introduce new columns or omit existing columns, it is recommended to avoid them as well.
 - It is not recommended to change the attributes of a table, such as table name and column name, using an SQL query. This action impacts the functioning of other saved objects in Table Editor.
4. Click the **Run Query** icon on the toolbar. The query results are displayed in the **Results** tab.

Specifying row conditions for tables

If you are using QMF for Workstation, you can specify row conditions to limit the rows that are displayed in the table. Without row conditions, all existing rows will be displayed. You can specify row conditions to limit the rows that are displayed in the table. Without row conditions, all existing rows will be displayed.

Related tasks

[Specifying sort conditions for tables](#)

You can specify the sort conditions that determine the order that the rows are displayed in tables.

[Editing database tables using Db2 special registers](#)

You can change data in database tables using Db2 special registers. A special register is a storage area that is defined for an application process by Db2 and is used to store information for the reference. A reference to a special register is a reference to a value provided by the current server. The list of available special registers depends on the version of the database.

[Copying tables](#)

Use the **Repository Explorer** view to copy tables inside the currently open data source.

Using Prompted tab

About this task

You use the **Row conditions** pane of the **Prompted** menu to specify the row conditions for tables. The **Prompted** menu of the **Table** editor is similar to the **Prompted Query** editor, except that only the **Sort conditions** and **Row conditions** panes are available.

Procedure

1. Right-click a table in the **Workspaces** view and select **Open With > Table Editor**.
The table opens in the editor window.
2. Click the **Prompted** tab.
The **Prompted** menu opens with only the **Sort conditions** and **Row conditions** panes available.
3. Click **Add Row Condition** in the **Row conditions** toolbar.
The **Row Conditions** window opens.
4. If you are specifying more than one condition that must be met for a row to be included in the table, you must indicate how each condition will be connected using the **Connector** radio buttons. If you select **And**, then the current condition that you are specifying and the previous condition (listed in the **Row conditions** pane of the **Prompted** menu) must be met in order for the row to be included. If you select the **Or**, then the current condition or the previous condition can be met in order for the row to be included. When you connect multiple conditions using both **And** and **Or**, then the rules of the data source determine how the conditions are processed.
5. Select a column from the **Select a column from the list** list box to examine for inclusion in the table.
The **Select a column from the list** list box lists all the columns that will be included in the table. To specify a conditional expression that will determine the column, enter an expression in the **Or, enter an expression here** field. Click the ellipsis (...) to open an expression builder that will help you build the conditional expression.
6. Specify the operator for the row condition expression using the **Operator** field.
To use the operator as is, select **Is**. To change the operator to its negative opposite, click the **Is Not**. For example, to specify "Is Not Equal To" select **Is Not** and select the **Equal To** operator in the list. You cannot use the **Starting with**, **Ending with**, or **Containing** operators if the left and right sides of the row condition expression have numeric data types. You cannot use any of the spatial predicates unless the left and right sides have spatial data types.
Note: If IBM Db2 Spatial Extender is installed on the database, you are able to work with spatial data types: points, lines, and polygons. But you cannot use any of the spatial predicates unless the left and right sides have spatial data types.
7. Specify the condition for which to check the rows in the **Right Side** area.

- a) You can use the expression builder button to help you build the conditions.
 - b) To add row conditions from a file of any format with the values that are separated by commas, click **Import Conditions From File**.
 - c) To specify row conditions from a set of query results, click **Import Conditions via Executing SQL Query**, and enter the SQL text of a query.
 - d) You can use the **Remove Condition** or **Remove All Conditions** buttons to remove one or all row conditions.
8. Click **Add**.
- The row condition is added and displayed in the **Row conditions** pane of the **Prompted** menu. Repeat this process for each condition that you want to add. When you have finished specifying each condition, click **Close**. The **Row Conditions** window closes. You are returned to the **Prompted** menu.

Using SQL tab

Procedure

1. Right-click a table in the **Workspaces** view and select **Open With > Table Editor**.
The table opens in the editor window.
2. Click the **SQL** tab.
The **SQL** tab opens displaying the SQL query that fetches the table displayed in the **Results** tab.
3. Specify the row conditions that will be used to filter the query results:
 - Use the WHERE clause to specify the filter conditions.
 - Use the AND, OR or NOT operators to combine multiple filter conditions.

For example:

```
SELECT [Columns] FROM [Table_Name]
WHERE [Condition1] AND [Condition2] AND [Condition2]...
```

Note: Consider the following when specifying an SQL query in the **SQL** tab:

- Use an SQL query to fetch the contents of a single table only. You will not be able to use the Table Editor to update the resultant table if you query more than one table in your SQL queries. Use separate table editors to query and update tables individually.
 - Ensure that the SQL query that you specify fetches all the columns of the table. If all the columns of a table are not included in the result set, you may not be able to effectively use the Table Editor to update the table. You may encounter errors when you try to insert new rows into such tables. Therefore, it is recommended to avoid introducing new columns or omitting existing columns in the SQL query. Because joins, aggregate functions, and calculated columns may introduce new columns or omit existing columns, it is recommended to avoid them as well.
 - It is not recommended to change the attributes of a table, such as table name and column name, using an SQL query. This action impacts the functioning of other saved objects in Table Editor.
4. Click the **Run Query** icon on the toolbar. The query results are displayed in the **Results** tab.

Editing database tables using Db2 special registers

You can change data in database tables using Db2 special registers. A special register is a storage area that is defined for an application process by Db2 and is used to store information for the reference. A reference to a special register is a reference to a value provided by the current server. The list of available special registers depends on the version of the database.

Before you begin

Open the database table that you want to edit on the **Results** tab of the Table Editor.

About this task

To edit a database table using special registers:

Procedure

1. Enter a special register to a table cell, which value you want to edit.
2. Commit the changes to replace the special registers that you entered by the values of the special registers. See the [“Working with database tables” on page 117](#) topic for the detailed information on editing, inserting data, and committing changes to the tables.

Note: If you use the special register to change the column value in one of the duplicate rows of the table, the column values are updated in all duplicate rows.

The database casts the data type of the special register value to the column data type.

Related tasks

[Specifying sort conditions for tables](#)

You can specify the sort conditions that determine the order that the rows are displayed in tables.

[Specifying row conditions for tables](#)

If you are using QMF for Workstation, you can specify row conditions to limit the rows that are displayed in the table. Without row conditions, all existing rows will be displayed. You can specify row conditions to limit the rows that are displayed in the table. Without row conditions, all existing rows will be displayed.

[Copying tables](#)

Use the **Repository Explorer** view to copy tables inside the currently open data source.

Copying tables

Use the **Repository Explorer** view to copy tables inside the currently open data source.

About this task

To copy a table, complete the following procedure:

Procedure

1. In the **Repository Explorer** view, right-click the table that you want to copy and select **Copy** from the menu.
2. Right-click the **Tables** node of the tree of objects and select **Paste** from the menu.
The **Copy Table** window opens.
3. In the **Destination Table Parameters** area, type the name and the owner for the new table in the **Table name** and **Table owner** fields.
4. In the **Method** area, specify the method of saving the data:
 - If you select **Regular**, QMF sends the query results back to the database and inserts them into the database table by using an individual SQL INSERT statement for each row of data.
 - If you select **Fast**, QMF adds SQL to the original query, reruns the query, and saves the data directly into the specified table.
5. In the **Existing Data Option** area, specify how existing data is to be handled:
 - **Replace any existing data** to replace any existing data that is in the specified table on the database.
 - **Add to any existing data** to add to any existing data that is in the specified table on the database.
6. In the **Commit scope** field, specify the number of rows to insert before committing the changes. If this value is set to zero, all of the rows are inserted before a commit occurs.
7. Click **OK**.

Related tasks

[Specifying sort conditions for tables](#)

You can specify the sort conditions that determine the order that the rows are displayed in tables.

Specifying row conditions for tables

If you are using QMF for Workstation, you can specify row conditions to limit the rows that are displayed in the table. Without row conditions, all existing rows will be displayed. You can specify row conditions to limit the rows that are displayed in the table. Without row conditions, all existing rows will be displayed.

Editing database tables using Db2 special registers

You can change data in database tables using Db2 special registers. A special register is a storage area that is defined for an application process by Db2 and is used to store information for the reference. A reference to a special register is a reference to a value provided by the current server. The list of available special registers depends on the version of the database.

Chapter 5. Selecting a new data source

You use the **Set Data Source** dialog to select a data source against which a query object will be run.

About this task

All query objects are associated with a default data source. When query objects are run, they are automatically run against the default data source. The default data source for a query object is the last used data source before the query was saved or in the case of new queries that have not yet been saved, the default is the data source last used in the QMF session.

To select the data source where you want to run the current query object:

Procedure

1. Open the **Set Data Source** window by selecting the **Set Data Source** command from one of the following menus:
 - Query
 - Form
 - Procedure
2. Select the data source where you want to run the query object from the **Data source** list. The **Data source** list includes all the data sources that are available for your use based on your user ID.
3. In the **User name** field, type the user ID that you want QMF to use to connect to the new data source.
4. In the **Password** field, specify the password associated with the user ID that you typed in the **User name** field.
5. To save the password that you have specified for this user ID and use it every time to connect to this data source, select the **Remember this password** check box.
6. Click **OK**. The object is run against the data source that you have specified in the **Data Source** field.

Chapter 6. Setting fonts for SQL statements

If you are using QMF for Workstation, you can set unique font attributes for the SQL statement text.

About this task

To set unique font attributes:

Procedure

1. Display SQL statements in the workstation query editor.
2. Open a the **Font** window one of the following ways:
 - Select **Query > Set Font**.
 - Select **Form > Set Font**.
 - Select **Procedure > Set Font**.
3. Select the font for the SQL statements that are displayed in the query window from the list of available fonts in the **Font** field.
4. In the **Font style** field, specify the style of the font.
5. In the **Size** field specify the size of the font.
6. Click **OK** to set the new font settings that you have specified. The **Font** window closes.

Chapter 7. Specifying substitution variables

Substitution variables are used to enter changing values to a SQL query at run time.

About this task

This feature enables you to substitute a part of an SQL statement and make it more generic. Substitution variables are active only while the object (query, form, or procedure) is running. As a result, only one object can access the substitution variable. The variable will not exist after the object is executed.

A substitution variable is special text in a query that begins with an ampersand character (&). A substitution variable can contain up to 18 alphabetic, numeric or special characters.

A substitution variable can appear anywhere in a query. The value of the substitution variable can be anything used in a query (except a comment). For example, you can use a substitution variable in place of a column name, search condition, subquery, or any specific value.

In the following example, you will be prompted for a customer number each time you run the following query:

```
SELECT ORDERNO, SALESREPNO, PRODNO, QUANTITY, &CUSTNO AS CUSTOMER#  
FROM Q.SALES
```

When you run the query and supply customer number at the prompt, the query will retrieve only those records that are associated with the specified customer number. Later you can launch the query and provide a different customer instead of writing a separate query.

To use a substitution variable:

Procedure

1. Open a query.
2. Type this SQL statement: `SELECT * FROM Q.STAFF WHERE DEPT = &MIN_DEPT`
3. Run the query. The **Prompt Variables** window opens.
4. Type 50 in the **Value** field of the window.
5. Click **OK**.

The query will run with the value 50 for DEPT.

What to do next

Try experimenting with substitution variables by replacing values in the **SELECT** and **FROM** clauses. See what results your queries return.

Chapter 8. Working with procedures

A procedure is a set of commands that enable you to run queries, print reports, import and export data, as well as perform other functions.

Use the **Procedure** window to create, open, display, and run procedures. A procedure is a set of QMF procedure commands that enable you to run queries, print reports, import and export data, as well as perform other functions. Procedures can be saved in a repository, in a QMF catalog, or to a file. All commands issued through procedures are governed by your resource limits. Procedures can contain:

- any QMF procedure command
- comment lines
- blank lines
- RUN commands that run other procedures or queries
- substitution variables

Note:

- QMF supports legacy job files. You must convert the legacy job files before they can be run. For Windows operating systems, you can schedule converted job files to run using the Windows Scheduler. You can also run converted job files from the QMF for Workstation command line.
- QMF supports procedures with logic (REXX procedures). Procedures with logic include REXX language directives. Users must have OBJECTREXX installed to run procedures with logic.

Restriction: REXX procedures are not supported in the 64-bit application.

Creating a new procedure

You can use the **Procedure** window to create new procedures.

About this task

To create a procedure:

Procedure

1. From the **User** perspective, select **File > New > Procedure**. The **Create Procedure** wizard opens. You can also select the **New Procedure** toolbar button.
2. Specify a name for the new procedure in the **Name** field.
3. Select from the list of available data sources in the **Data source** field where the new procedure will reside if saved. If you leave this field blank, the new procedure will be saved on the current data source.
4. Click **Finish**. The new procedure object is opened in the editor view of the **User** perspective.
5. In the **Procedure** editor, enter the commands that you want to include in the procedure. At any time while typing the procedure commands, you can press **Ctrl+Spacebar** to access the Content Assist feature. For more information about the Content Assist feature, see the [“Using Content Assist for visual queries”](#) on page 59 topic.

If you are creating a procedure with logic using REXX language directives, you must type a REXX comment line as the first line of the procedure. The REXX comment line should appear as `/*REXX*/`. In addition, if you are including any QMF procedure commands in the procedure with logic, the QMF procedure commands must be enclosed in single quotes.

Restriction: REXX procedures are not supported in the 64-bit application.

6. Save, run or print the procedure. After running a procedure, results are returned in the available editor window. If you are running a procedure with logic, communication with the procedure is through the **REXX Console** view.
7. You can schedule a procedure to run at another time using the QMF for WebSphere task scheduler or your local scheduler.

Related concepts

[JavaScript functions in procedures](#)

The following topics describe the functions for the proc object and the result set of the current open query.

Related tasks

[Creating JavaScript procedures](#)

You can use JavaScript syntax to create more complex and not linear procedures.

[Creating procedures as flowcharts](#)

You can use the enhanced procedure editor to create linear or complex JavaScript procedures as flowcharts.

Related reference

[Procedure commands](#)

In this section, each procedure command is described. Descriptions include syntax diagrams and usage examples.

Creating JavaScript procedures

You can use JavaScript syntax to create more complex and not linear procedures.

About this task

To create a JavaScript procedure:

Procedure

1. In the **Procedure** editor, type `/*JavaScript*/`.
The current procedure is processed as a JavaScript one.
2. Type a script for the JavaScript procedure in the **Procedure** editor. Use the proc object and its functions.
Note: JavaScript procedures support the syntax of JavaScript modules.
3. Save or run the created procedure.

Related concepts

[JavaScript functions in procedures](#)

The following topics describe the functions for the proc object and the result set of the current open query.

Related tasks

[Creating a new procedure](#)

You can use the **Procedure** window to create new procedures.

[Creating procedures as flowcharts](#)

You can use the enhanced procedure editor to create linear or complex JavaScript procedures as flowcharts.

Related reference

[Procedure commands](#)

In this section, each procedure command is described. Descriptions include syntax diagrams and usage examples.

Creating procedures as flowcharts

You can use the enhanced procedure editor to create linear or complex JavaScript procedures as flowcharts.

About this task

To add a command to a flowchart procedure:

Procedure

1. Open the procedure editor and click the **Design** tab.
The **Procedure Designer** opens.
2. To add a procedure command, double-click the block on the **Design** tab. You can also drag the command that you want to add from the **Palette** view to the block.
3. In the **Add Command** wizard from the **Command** list, select a command, which you want to add to the procedure.
4. For some commands select an object, to which you want to apply the command, from the **Objects** list.
5. If the command with the specified object requires parameters, click **Next**, and specify them on the second page of the **Add Command** wizard:
 - a) In the **[command_name] Command** area, enter required values to corresponding fields. You can click **Browse** to search for the object that you want to add or specify the directories that you want to use in commands.

Tip: Use the **Browse** button to add double quotation marks to the object name automatically. If you enter an object name manually, do not forget to add double quotation marks to it.

- b) For some commands, select one of the following options:
 - **Current object** to use the object that is opened in the application.
 - **Object** to use the object that is saved in the repository or QMF catalog.
 - **JavaScript variable** to use a previously created JavaScript variable. See an example of a JavaScript variable and its use in a JavaScript procedure:

Example

JavaScript variable:

```
var tableName = proc.prompt("Enter a table name", "");
```

JavaScript procedure:

```
/*JavaScript*/  
var tableName = proc.prompt("Enter a table name", "");  
proc.exec('DISPLAY TABLE ' + tableName);
```

- c) Specify the command parameters in the **Parameters** area. For more information about available procedures and their parameters, see the [Procedure commands](#) section.
6. Click **Finish** to close the wizard and add the command to the procedure. Follow these steps to add each command to the procedure.

Tip: Use **Comment** option to add comments to the commands in the flowchart. They are visible only in the **Procedure Designer**.

Related concepts

[JavaScript functions in procedures](#)

The following topics describe the functions for the proc object and the result set of the current open query.

Related tasks

[Creating a new procedure](#)

You can use the **Procedure** window to create new procedures.

[Creating JavaScript procedures](#)

You can use JavaScript syntax to create more complex and not linear procedures.

Related reference

[Procedure commands](#)

In this section, each procedure command is described. Descriptions include syntax diagrams and usage examples.

Editing flowchart procedures

You can edit existing procedure commands in the **Procedure Designer**.

About this task

To edit a flowchart procedure:

Procedure

1. Open the procedure that you want to edit on the **Design** tab.
2. Double-click the command that you want to edit. The **Edit Command** wizard opens.
3. Add the changes that you want to apply to the command.
Tip: To replace any command with a new one, use the **Edit Command** wizard.
4. To add a command before or after an existing command, drag it from the **Palette** to the command before or after the existing command.
 - a) Specify the command position on the first page of the **Add Command** wizard. Click **Next**
 - b) On the second page of the wizard, specify the command and its parameters. Click **Finish**.
5. You can also modify the appearance of the procedure flowchart by doing the following actions:
 - To add all commands one after another by default, right-click the **Design** tab and select **Auto Layout**.
 - To add commands in the custom order, right-click the **Design** tab and clear **Auto Layout**.
 - To edit the location of the procedure block, drag it to a new position or right-click the block and select the **Up** and **Down** items.
 - To edit the alignment of the procedure blocks, right-click the required block, select **Align**, and specify the alignment that you want to apply to the current block.
 - To match the procedure blocks by height or width, right-click the block, select **Match**, and specify the option that you want to apply to the current block.

Adding JavaScript conditions to procedures

You can add JavaScript conditions to procedures to make them complex and not linear.

About this task

To add a JavaScript condition in the **Procedure Designer**:

Procedure

1. Open the procedure to that you want to add a condition on the **Design** tab.
2. From the **Palette**, select the **Condition** option and drag it to the empty block of the procedure chart.

3. In the **JavaScript Condition** window, type the condition that you want to include in the procedure.
4. Click **OK** to add the condition.

Results

- Under the **Yes** arrow, specify the command or condition that you want to apply if the first condition is met.
- Under the **No** arrow, specify the command or condition that you want to apply if the first condition is not met.

Procedure commands

In this section, each procedure command is described. Descriptions include syntax diagrams and usage examples.

Related concepts

[JavaScript functions in procedures](#)

The following topics describe the functions for the `proc` object and the result set of the current open query.

Related tasks

[Creating a new procedure](#)

You can use the **Procedure** window to create new procedures.

[Creating JavaScript procedures](#)

You can use JavaScript syntax to create more complex and not linear procedures.

[Creating procedures as flowcharts](#)

You can use the enhanced procedure editor to create linear or complex JavaScript procedures as flowcharts.

Interface differences

Each QMF interface supports a slightly different list of procedure commands and procedure command parameters.

Users should take these differences into consideration if they are writing procedures using one interface and plan on running the procedure using a different interface. Some commands and parameters will run on one interface and will not run using another interface. Each interface documents the commands and parameters that are currently supported.

Typically, when an unsupported procedure command is encountered, the interface will issue an error message and stop running the procedure.

In some instances, the interface will ignore the unsupported procedure command or procedure command parameter. When this happens, the interface will not issue an error message and will continue running the procedure. The unsupported procedure command or procedure command parameter is skipped. Depending on the functionality the results of running such a procedure can be incorrect. However, by not stopping the procedure and issuing an error message, some procedures developed using one interface can be run using another interface. The commands and command parameters that are ignored are noted in the interface's documentation.

Procedure syntax

You can find a summary of the syntax rules that are common to each command in this topic.

Typically, each line of a procedure contains a single command. However, you can spread a command over multiple lines by placing a plus sign (+) as a continuation character in the first column of each additional line.

You can include comments in a procedure by entering two hyphens (--) on a line. All text that follows the hyphens through the end of the line is a comment and is not interpreted in the procedure.

You can use substitution variables to parameterize a procedure. Each time that you run a procedure; it is scanned for substitution variables before it is run. You must provide values for all substitution variables before the procedure is run. Values for substitution variables can come from the RUN PROC command, from global variables, or by displaying the **Enter Substitution Variable Values** window.

Procedure parameters

Typically, each procedure parameter has the following syntax: <parameter_name> = <parameter_value>.

The procedure parameters must be separated by commas or spaces. See the following commands that are identical:

- RUN QUERY Q.STAFF (CONFIRM = NO ROWLIMIT=10 &&ID=10
- RUN QUERY Q.STAFF (CONFIRM = NO, ROWLIMIT=10, &&ID=10
- RUN QUERY Q.STAFF (CONFIRM=NO,ROWLIMIT=10,&&ID=10

If you want to use a parameter value, which contains commas or spaces within, you must put the parameter in single ('parameter value') or double quotation marks ("parameter value") or put it in parentheses ((parameter value)).

Note: If you put the parameter value in single or double quotation marks, these marks are included in the parameter value. If you do not need quotation marks in the parameter value, put it in parentheses.

The following commands have different syntax in parameter values, and they are processed differently by the application:

- RUN QUERY Q.STAFF (&&NAME="MY NAME" - In this procedure, the parameter value is processed as "MY NAME" with double quotation marks.
- RUN QUERY Q.STAFF (&&NAME='MY NAME' - In this procedure, the parameter value is processed as 'MY NAME' with single quotation marks.
- RUN QUERY Q.STAFF (&&NAME=(MY NAME) - In this procedure, the parameter value is processed as MY NAME without characters.

Reading the syntax diagrams

Syntax diagrams provide details on how to write specific procedure commands.

The procedure command syntax diagrams read from left to right, top to bottom. Each command starts with ">>" and ends with "<<". If a command is continued from one line to the next, the first line ends with "<" and the next line begins with ">".

A command can have two types of parameters. Positional parameters must be placed in a certain position within a command. Keyword parameters are assigned a value and can be placed in any order within a command. The first keyword parameter used in a command must be preceded by a left parenthesis. No parameter can be longer than 80 characters. All parameters are separated from each other with a comma. A right parenthesis is not required, but can be used to end the command.

Required Parameters

Parameters are on the main path if they are required.

```
>>-- CONNECT TO ServerName -----<<
```

When one parameter is on the main path with others listed on a separate line, you must select at least one from parameter from the list.

```
>>-- DISPLAY --- QUERY -----<<
      +- PROC ----+
      +- FORM ----+
      +- REPORT --+
```

Optional Parameters

When a parameter is part of a list that is separate from the main path, it is optional. When all parameters are shown in a list separate from the main path, you can specify any one of them or none.

```
>>-- ERASE --- ObjectName -----<<
      +- QUERY  -+
      +- FORM   -+
      +- PROC   -+
      +- TABLE -+
```

Sometimes two values are separated by a forward slash (/). This indicates that one of the two values must be entered:

```
>>-----<<
      +- ( CONFIRM = YES/NO -+
```

QMF Catalog objects versus Repository folder objects

When using procedure commands that manipulate QMF objects, (queries, forms, visual reports, or procedures) different syntax should be used for identifying objects that are stored in a QMF Catalog and objects that are stored in a Repository folder.

Objects stored in QMF Catalogs

When QMF objects are stored in a QMF Catalog, the syntax for referring to them in a procedure is to refer to them by <TYPE> <OWNER>.<OBJECT_NAME>. For example:

Example 1:

```
RUN QUERY ADMIN.TESTQ
DISPLAY FORM ADMIN.TESTF
```

In the QMF Catalog, <OWNER> .<OBJECT_NAME> is unique, so the <TYPE> identifying is optional but recommended for making the procedure readable. So, this procedure:

Example 2:

```
RUN QUERY ADMIN.TESTQ
DISPLAY FORM ADMIN.TESTF
```

could also be written as:

```
RUN ADMIN.TESTQ
DISPLAY ADMIN.TESTF
```

Also, if the procedure that is manipulating the QMF Catalog object is being run by the owner of the object, then the <OWNER> identifying is also optional. For example, if the procedure is being run by someone who is signed on as ADMIN, then it could also be written as follows:

Example 3:

```
RUN QUERY TESTQ
DISPLAY FORM TESTF
```

or

```
RUN TESTQ
DISPLAY TESTF
```

While the shorter object identifiers are supported and are useful in certain situations, in general it is best practice to fully qualify the procedure object identifiers as shown in [Example 1](#).

Note: In general procedure references to QMF Catalog names are not case sensitive. It is possible but not common for QMF Catalog objects to have more exotic names than the traditional eight character names. For example, mixed case names and names with spaces and special characters are supported if during

the initial save the name is enclosed in double quotes. Thereafter all references to the object must also encase the object name in double quotes. For example:

Example 4:

```
RUN QUERY ADMIN."Test Case for Query @ HQ"
```

Objects stored in Repository folders

When QMF objects are stored in a Repository folder, the syntax for referring to them in a procedure is to refer to them by <TYPE> <OBJECT_NAME>. For example:

Example 5:

```
RUN QUERY TESTQ
```

Notice that the syntax of this example matches the syntax of shown in [Example 3](#) in the QMF Catalog section. Therefore when a procedure command is submitted there is a search order that QMF uses to locate the objects referenced in the procedure.

1. QMF looks for a QMF Catalog and searches it for a matching <OWNER> . <OBJECT_NAME>. If one is found it is used. If no <OWNER> is supplied as in the [Example 5](#), then the current user ID is used for the <OWNER> value as described in the third example of the QMF Catalog section. This means that procedures that are stored in a Repository folder can access objects that are stored in the QMF Catalog.
2. QMF searches the folder in the Repository containing the procedure that is being run. When authoring a new procedure from scratch, this is very important to keep in mind. The new procedure does not have a Repository folder until it has been saved to the Repository. QMF will search the QMF Catalog and if it does not find a match then the search stops and the Repository objects will not be found.
3. QMF searches the rest of the Repository folders for a match. This search will not take place if the procedure has not been saved to a Repository folder yet. Caution: The result of the search among the rest of the Repository tree might not result as expected if there is more than one object of the same name. Unlike the QMF Catalog, the Repository tree does not enforce uniqueness on object names outside the same folder. Therefore, for procedures that refer to objects by <OBJECT_NAME> , it is highly recommended that the objects be in the same folders as the procedure that references them.

Using the Key property to reference objects

With a complex Repository tree and objects that are stored outside the same folder as the procedure that references them, another syntax is available that guarantees uniqueness. When an object is selected in the Repository tree, the **Properties** view for that object contains a field called **Key**. The Key is unique and can be used by a procedure to reference the object. For example:

```
RUN QUERY qmf:/.workspaces/Default/Queries/TESTQ
```

If text in the Key contains spaces, then the Key must be enclosed in double quotes. Procedures in development that have not been saved to the Repository yet can be run using the Key syntax. A procedure that is stored in the QMF Catalog can use the Key syntax to access objects that are stored in Repository folders.

BOTTOM command

The BOTTOM command scrolls to the last row of a query result set.

This command is equivalent to FORWARD MAX.

Syntax

```
>>-- BOTTOM -----<<
```

Example

```
BOTTOM
```

CONNECT command

The CONNECT command establishes a connection to a database data source.

Subsequent procedure commands will run at the specified data source. The running procedure's server will also be changed to the specified server. No immediate action is taken upon any other current objects within the procedure. However, subsequent commands which affect those objects might result in additional processing.

For example, assume query Q1 has been run but not completed at server X. The CONNECT TO 'SERVER Y' command is then issued, connecting to server Y. Next, the SAVE DATA AS T1 command is issued.

Query Q1 is not immediately affected by the CONNECT command to server Y, but the SAVE DATA command requires completion (or cancellation) of query Q1 at server X. Once the data object is completed, the server for query Q1 is changed to Y, and the data is saved in table T1 at server Y.

Syntax

```
>>-- CONNECT ----- TO ServerName -----<<
      +- UserName -+                (PASSWORD=password)
```

| Parameter | Description |
|------------|---|
| ServerName | The name of a database server. If the name includes spaces, enclose it in quotes. |
| UserName | The user name that will be used to connect to the database server. UserName is an optional parameter. |
| PASSWORD | The password that will be used to connect to the database server. If you specify UserName, you must supply a password using the PASSWORD parameter. |

Example

```
CONNECT TO SERVER1
```

```
CONNECT pjones TO SERVER1 (PASSWORD=sales)
```

CONVERT command

The CONVERT command will convert the current query (viewed in either SQL or Query Builder editor) to the SQL view.

The original query (whether a named object in the database or a temporary object) is unaffected by this operation.

Syntax

To convert a query that is stored in a database:

```
>>-- CONVERT ----- QueryName -----<<
      +- QUERY -+                (+- SUBSTITUTE = YES/NO --+)
```

To convert a query currently in an open window (in temporary storage):

```
>>-- CONVERT QUERY -----<<
```

| Parameter | Description |
|------------|--|
| QueryName | The name of the query that is to be converted. |
| SUBSTITUTE | Defines whether substitution should take place for query variables. Available options are: <ul style="list-style-type: none"> • YES. If you have variables in your query, QMF attempts to substitute values for them. If all the variables are defined (either with the &variable parameter or predefined global variables), no prompt panel is displayed. If QMF cannot resolve all the variables, it prompts you to enter required values. • NO. No variable names in your query are resolved. |

Example

The following example converts the query that is in an open window (in temporary storage):

```
CONVERT QUERY (SUBSTITUTE=YES
```

The following examples convert the Q.STAFF query that is stored in QMF Catalog:

```
CONVERT Q.STAFF
```

```
CONVERT QUERY Q.STAFF
```

DISPLAY command

The DISPLAY command displays an object derived from the content in an open window (such as query results), or an object saved in the repository or QMF catalog.

Syntax

To display an object:

Note:

To display an object derived from the content in the last open window:

```
>>-- DISPLAY ----- ObjectName -----<<
+- QUERY -----+
+- FORM -----+
+- .MAIN-----+
+- .BREAK1-----+
+- .BREAK2-----+
+- .BREAK3-----+
+- .BREAK4-----+
+- .BREAK5-----+
+- .BREAK6-----+
+- .CALC-----+
+- .COLUMNS-----+
+- .CONDITIONS--+
+- .DETAIL-----+
+- .FINAL-----+
+- .OPTIONS-----+
+- .PAGE-----+
+- PROC -----+
+- REPORT -----+
```

In the DISPLAY command, you can provide either the object name or the full object key of the object being displayed. If the procedure is stored in a QMF catalog, objects stored in the same catalog can be displayed using only their object name. For example, if the Q.STAFF query is stored in the same catalog as the procedure, you would use DISPLAY Q.STAFF to display the Q.STAFF query.

Note: The DISPLAY FORM.*Section#* command either activates a section of the form structure tree in the **Classic Report** editor if the form is opened, or it creates a new form if one does not yet exist.

If the procedure is stored in the repository, objects stored along side the procedure in the same repository folder can be displayed using only their object names. For example, if the Q.STAFF query is stored in the same repository folder as the procedure, then you could use DISPLAY Q.STAFF to display the Q.STAFF query.

However, if the object to be displayed is not stored in the procedure, nor in the same repository folder, then you must supply the full object key in the DISPLAY command. For example, DISPLAY "rsbi:/.workspaces/Traditional DBA View/Query1".

Note: An object's key can be viewed by clicking the object in the repository or workspace explorer and reviewing the key property value in the **Properties** view. If the key contains spaces, make sure it is enclosed by double quotes. To display the current object in visual report format, you must use the **SHOW** command. If you are displaying a visual dashboard, you must supply the full object key, since visual dashboards can not be stored in QMF catalogs.

| Parameter | Description |
|------------|--|
| ObjectName | The name or the full object key of the object to be displayed. |
| &&Variable | If the object is a query or procedure, assigns a value to a variable in the query or procedure that is run. The variable name can be from 1 to 17 characters long and the value can be from 1 to 55 characters long. You can specify any number of variables and values on the DISPLAY command. If there are variables in the query or procedure that are not given values on the DISPLAY command, and are not global variables, the user will be prompted for the values. When you include variable assignments in a procedure, you must use two ampersands to prevent variable substitution before the procedure is run. |

Examples

The following example displays the query that is stored in the QMF Catalog:

```
DISPLAY USER1.QUERY1
```

The following example displays the query that is stored in the repository:

```
DISPLAY "rsbi:/.workspaces/Traditional DBA View/Query1"
```

```
DISPLAY QUERY
```

```
DISPLAY FORM.MAIN
```

DRAW command

The DRAW command creates a basic query for a table based on the description of the table in the database.

Syntax

```
>>-- DRAW TableName -----<<
      ( +- TYPE = SELECT/INSERT/UPDATE +-
        +- IDENTIFIER = CorrName      +-)
```

| Parameter | Description |
|------------|---|
| TableName | The name of the table for which to create a query. |
| TYPE | Specifies the type of SQL query to create. The default is SELECT. |
| IDENTIFIER | The correlation name to be associated with the table in the resulting query. It is ignored when TYPE=INSERT. There is no default. |

Examples

```
DRAW Q.STAFF (TYPE=SELECT
```

EDIT command

The EDIT command displays an object derived from the content in an open editor window (such as queries, forms, procedures, and tables), or an object saved in the repository or QMF catalog.

Syntax

To edit an object in the editor window:

To edit an object derived from the content in the last open window:

```
>>-- EDIT -----+- QUERY --+-----<<
                    +- FORM ---+
                    +- PROC ---+
                    +- REPORT -+
```

In the EDIT command, provide the full object key of the object that you want to edit.

Note: The object key can be viewed by clicking the object in the repository or workspace explorer and reviewing the key property value in the **Properties** view. If the key contains spaces, make sure to place it within quotes. To display the current object in visual report format, you must use the [SHOW](#) command.

| Parameter | Description |
|------------|--|
| ObjectName | The full object key of the object to be edited. |
| &&Variable | If the object is a query or procedure, assigns a value to a variable in the query or procedure that is run. The variable name can be from 1 to 17 characters long and the value can be from 1 to 55 characters long. You can specify any number of variables and values on the EDIT command. If there are variables in the query or procedure that are not given values on the EDIT command, and are not global variables, the user will be prompted for the values. When you include variable assignments in a procedure, you must use two ampersands to prevent variable substitution before the procedure is run. |

Examples

The following example edits the query that is stored in repository:

```
EDIT "qmf:/.workspaces/Traditional DBA View/Query1"
```

The following example edits the object that is derived from the content in the last open window

```
EDIT QUERY
```

ERASE command

The ERASE command removes an object (query, form, procedure, or table) from the database.

Syntax

| Parameter | Description |
|------------|---|
| ObjectName | The name of the object to remove from the database. |

| Parameter | Description |
|-----------|---|
| CONFIRM | Specifies whether or not to display a confirmation dialog before removing the object. If CONFIRM is not specified or is NO, the corresponding resource limit is used. |
| FOLDER | Specifies the folder in QMF Catalog that stores links to the objects that you want to erase. The default value is provided by the DSQEC_CURR_FOLDER global variable. Note: If the folder is specified, only a link to the object is removed from the folder and the object remains in QMF Catalog. If the folder is not specified, all links to the object are removed from all folders and the object itself is removed from the database. |

Example

The following example erases the query that is stored in QMF Catalog.

```
ERASE QUERY USER1.QUERY1
```

EXECUTE command

The EXECUTE command activates the target application, object, or URL.

Syntax

The command is similar to WINDOWS command.

Note: This command is ignored in QMF for WebSphere.

```
>>-- Execute CommandLine -----<<
```

| Parameter | Description |
|-------------|---|
| CommandLine | An application command line, object name, or URL to launch. The command is launched and execution of the procedure continues without waiting for completion of the command. |

Example

```
EXECUTE c:\programs\notepad.exe
```

EXPORT command

The EXPORT command saves objects that are stored in a database or are currently in an open window to a file.

Syntax

To export an object from a database to a file:

```
>>-- EXPORT --- QUERY --- ObjectName TO FileName -----<<
      +- PROC ---                               ( +- CONFIRM = YES/NO +-
                                                +- SAVEATSERVER = YES/NO +-
```

```
>>-- EXPORT FORM ObjectName TO FileName -----<<
      ( +- CONFIRM = YES/NO +-
        +- LANGUAGE = ENGLISH/SESSION +-
        +- SAVEATSERVER = YES/NO -----+
```

```

>>-- EXPORT TABLE ObjectName TO FileName -----<<
      ( +- CONFIRM = YES/NO
      +- DATAFORMAT = CSV/DBF/HTML/IXF/PDF/SHP/TEXT/
WQML/XLS/XLSX/XML -+
      +- DATEFORMAT = Java date format string
      +- TIMEFORMAT = Java time format string
      +- OUTPUTMODE = BINARY/CHARACTER/PC
      +- LOBSINFILE = YES/NO
      +- LOBSTO = path1;path2;...
      +- LOBFILE = basefile1;basefile2;...
      +- CCSID = integer or Java encoding name
      +- COLUMNHEADINGS = YES/NO
      +- UNICODE = YES/NO
      +- SAVEATSERVER = YES/NO

```

Note: EXPORT TABLE always exports names.

Restriction: LOBSINFILE, LOBSTO, and LOBFILE parameters are available only for the IXF data format.

To export an object currently in an open window to a file:

```

>>-- EXPORT --- QUERY --- TO FileName -----<<
      +- PROC ---+      ( +- CONFIRM = YES/NO -+
      +- SAVEATSERVER = YES/NO -+

```

```

>>-- EXPORT FORM TO FileName -----<<
      ( +- CONFIRM = YES/NO -----+
      +- LANGUAGE = ENGLISH/SESSION -+
      +- SAVEATSERVER = YES/NO -----+

```

```

>>-- EXPORT DATA TO FileName -----<<
      ( +- CONFIRM = YES/NO -----+
      +- DATAFORMAT = CSV/DBF/HTML/IXF/PDF/SHP/TEXT/WQML/XLS/XLSX/XML -+
      +- DATEFORMAT = Java date format string-----+
      +- TIMEFORMAT = Java time format string-----+
      +- OUTPUTMODE = BINARY/CHARACTER/PC -----+
      +- LOBSINFILE = YES/NO -----+
      +- LOBSTO = path1;path2;... -----+
      +- LOBFILE = basefile1;basefile2;... -----+
      +- CCSID = integer or Java encoding name -----+
      +- COLUMNHEADINGS = YES/NO -----+
      +- UNICODE = YES/NO -----+
      +- MODE = GRID/RAW -----+
      +- SAVEATSERVER = YES/NO -----+

```

Restriction: LOBSINFILE, LOBSTO, and LOBFILE parameters are available only for the IXF data format.

```

>>-- EXPORT REPORT TO FileName -----<<
      ( +- CONFIRM = YES/NO -----+
      +- DATAFORMAT = TEXT/HTML/PDF -----+
      +- SPLIT = YES/NO -----+
      +- ORDER = 0/1/2 -----+
      +- WIDTH = integer -----+
      +- LENGTH = integer -----+
      +- UNITS = INCHES/CENTIMETERS/MILLIMETERS/POINTS--+
      +- USEFORMPS = YES/NO -----+
      +- SEPARATOR = any HTML text -----+
      +- CCSID = integer or Java encoding name -----+
      +- SAVEATSERVER = YES/NO -----+

```

Note: If several reports of different types are open in the editor area, the EXPORT REPORT command exports visual reports first.

| Parameter | Description |
|------------|--|
| ObjectName | The name of the object to export from the database. |
| FileName | The name of the previously created file to which the object will be exported or a temporary directory C:\Temp. |
| CONFIRM | Specifies whether or not to display a confirmation dialog before replacing an existing file. If CONFIRM is not specified or is NO, the corresponding resource limit is used. |
| LANGUAGE | Specifies whether a form is exported in English or in the current session language. A form that is exported in English can be run in any NLS session. A form exported in the session language can only be run in a session of the same language. The default value is provided by the DSQEC_FORM_LANG global variable. |
| DATAFORMAT | <p>Specifies the file format of the exported data. If you omit this parameter, the DSQQW_EXP_DT_FRMT global variable supplies the format to be used. You can specify in the DSQQW_EXP_DT_FRMT global variable:</p> <ul style="list-style-type: none"> • zero (0) for text format • two (2) for HTML format • three (3) for CSV format <p>Note: The separator is always a comma unless the locale uses decimal comma. In this case, a separator of semi-colon is used.</p> <ul style="list-style-type: none"> • four (4) for IXF • five (5) for dbase III files • six (6) for Shapefile format • seven (7) for WQML format • eight (8) for XML format • nine (9) for PDF format • ten (10) for XLS format • eleven (11) for XLSX format <p>If you specify IXF, the DSQQW_EXP_OUT_MDE global variable can be set to either zero (0) for character mode System/370 IXF or one (1) for PC/IXF. When exporting reports, HTML format adds the <PRE> tag, TEXT is for a text-based file, and PDF is for a PDF-based file.</p> |
| DATEFORMAT | <p>Specifies how the date is formatted in the HTML, CSV or TXT export file.</p> <p>Date formats are specified by Java date pattern strings. Within date pattern strings, unquoted letters from 'A' to 'Z' and from 'a' to 'z' are interpreted as pattern letters representing the components of a date string.</p> <p>Text can be quoted using single quotes (') to avoid interpretation.</p> <p>Note: If the format string includes spaces, enclose it in quotes. For more information about Java format strings, see the Java 2 SDK, Standard Edition Documentation.</p> |

| Parameter | Description |
|----------------|---|
| TIMEFORMAT | <p>Specifies how the time is formatted in the HTML, CSV or TXT export file.</p> <p>Time formats are specified by Java time pattern strings. Within time pattern strings, unquoted letters from 'A' to 'Z' and from 'a' to 'z' are interpreted as pattern letters representing the components of a time string.</p> <p>Text can be quoted using single quotes (') to avoid interpretation.</p> <p>Note: If the format string includes spaces, enclose it in quotes. For more information about Java format strings, see the Java 2 SDK, Standard Edition Documentation.</p> |
| OUTPUTMODE | Specifies the output format of the data. |
| LOBSINFILE | Specifies whether or not LOBs are included in the exported data. |
| LOBSTO | The location to save the LOBs. |
| LOBFILE | The base name of the exported LOBs. |
| CCSID | Specifies the code page (coded character set identification number) to use when saving the file. This value can either be an integer or the Java encoding name of the code page. All of the Java encoding names that are listed in the Export dialog are supported. Any encoding names that are supported by your JVM can be used as well. |
| COLUMNHEADINGS | Specifies whether or not to export column headers. It is only available for export to HTML, CSV or TEXT files. |
| UNICODE | Specifies whether graphic columns will be saved as UNICODE. This option is only applicable when saving data in IXF format. |
| SPLIT | Specifies whether the report will be split into pages. Use only when exporting visual reports. |
| ORDER | <p>Controls how the report will be split into pages. Use only when exporting visual reports.</p> <ul style="list-style-type: none"> • 0 - each visual report page will be exported into separate files. • 1 - a single output file contains visual report pages in "across then down" order. • 2 - a single output file contains visual report pages in "down then across" order. |
| WIDTH | <ul style="list-style-type: none"> • Specifies the width in units for a report page. <p>Note: Used for the both classic and visual reports.</p> <ul style="list-style-type: none"> • Specifies the width of an exported Analytics object in pixels. |
| LENGTH | <p>Specifies the length in units for a report page.</p> <p>Note: Used for the both classic and visual reports.</p> |
| UNITS | <p>Specifies the measurement units that are used in the WIDTH and LENGTH parameters.</p> <p>Note: Use only when exporting visual reports.</p> |
| USEFORMPS | Specifies whether or not to use the Windows form associated with the object. |

| Parameter | Description |
|--------------|---|
| SEPARATOR | <p>Specifies a string that will be used as a page separator. Use quotes if necessary.</p> <p>Note: Use only when exporting visual reports.</p> |
| MODE | <p>Specifies whether the query result data is saved with formatting and added calculated columns. You can specify one of the following values:</p> <ul style="list-style-type: none"> • GRID to specify that all of the data as it is currently formatted in the current query results will be saved. Any calculated columns that have been added to the query results are included. <p>This is the default value for the PDF, XLS, and XLSX formats.</p> <p>Note: MODE GRID exports labels if DSQDC_COL_LABELS is set to 1. MODE GRID exports names if DSQDC_COL_LABELS is set to 0.</p> <ul style="list-style-type: none"> • RAW to specify that all of the data in the current query results will be saved. Any formatting that has been applied to the data will not be saved. Any calculated columns that have been added to the query results will not be saved. <p>This is the default value for all formats other than PDF, XLS, and XLSX.</p> <p>Note: MODE RAW always exports names. This also applies when the MODE parameter is omitted.</p> <p>Note: The parameter is ignored if DATAFORMAT=XLS, XLSX.</p> |
| SAVEATSERVER | <p>Specifies whether or not to include the Root output directory in the export path of an object. This directory is set on Server-Side File System page of the Preferences window. In QMF for WebSphere you can specify only the path, lying in the root directory set by administrator. Otherwise, the export will be forbidden and an error will occur.</p> |
| HEIGHT | <p>Specifies the height of an exported Analytics object in pixels.</p> |

Example

The following example exports a query from the QMF Catalog to the previously created directory:

```
EXPORT QUERY USER1.QUERY TO C:\Queries\query1.vry
```

The following is an example of using DATEFORMAT and TIMEFORMAT parameters:

```
EXPORT DATA TO C:\Results\example.txt (DATEFORMAT="yyyy MM dd", TIMEFORMAT=HH:mm)
EXPORT TABLE Q.INTERVIEW TO "C:\Tables\interview.txt" (DATEFORMAT=yyyy/MM/dd,
TIMEFORMAT="HH mm")
```

The following is an example of using SAVEATSERVER parameter. Consider the case when the root output parameter is set to *C:\Temp* on the **Server-Side File System** preferences page:

```
EXPORT QUERY USER1.QUERY TO query1.vry (SAVEATSERVER=YES)
```

The result of the command will be a new file in *C:\Temp\query1.vry*

Related tasks

[Setting the directory location preference for objects generated by Export actions](#)

You can set preferences in QMF so that objects generated from various QMF actions are exported to a specific location on either the local file system or the server file system.

FORWARD command

The FORWARD command scrolls to the last row of a query result set.

The only acceptable parameter for this command is MAX, making it equivalent to the [BOTTOM](#) command.

Syntax

```
>>-- FORWARD MAX -----<<
```

Example

```
FORWARD MAX
```

IMPORT command

The IMPORT command imports objects that are stored in a file and opens them in an available window (temporary storage) or saves them in the current database.

Syntax

To import an object from a file, open it in an available window and save it in the current database:

```
>>-- IMPORT --- QUERY --- ObjectName FROM FileName -----<<
      +- PROC -+                                     ( +- CONFIRM = YES/NO -----+
                                                    +- COMMENT = Text -----+
                                                    +- SHARE = YES/NO -----+
```

```
>>-- IMPORT FORM ObjectName FROM FileName -----<<
      ( +- CONFIRM = YES/NO -----+
        +- COMMENT = text -----+
        +- LANGUAGE = ENGLISH/SESSION -----+
        +- SHARE = YES/NO -----+
```

```
>>-- IMPORT TABLE ObjectName FROM FileName -----<<
      ( +- CONFIRM = YES/NO -----+
        +- COMMENT = Text -----+
        +- ACTION = REPLACE/APPEND -----+
        +- ACCELERATOR = acceleratorname -----+
        +- ACCELERATORDATABASE = databasename ---+
        +- SPACE = tablespace/database.tablespace+
        +- SPACE DATABASE = database-----+
```

To import an object, query result data or LOB data from a file to an open window (temporary storage):

```
>>-- IMPORT --- QUERY --- FROM FileName -----<<
      +- PROC -+
```

```
>>-- IMPORT -- DATA --- FROM FileName -----<<
      ( +- LOBSFROM = path1;path2; -+
```

```
>>--IMPORT --- FORM --- FROM FileName -----<<
      +- REPORT -+      ( +- LANGUAGE = ENGLISH/SESSION -----+
```

To import visual reports created in QMF for Windows:

```
>>-- IMPORT -- REPORT ObjectName -- FROM FileName -----<<
      ( +- COMMENT = Text -----+
        +- CONFIRM = YES/NO -----+
        +- LANGUAGE = ENGLISH/SESSION -----+
```

| Parameter | Description |
|---------------------|---|
| ObjectName | The name of the object to import into the database. |
| FileName | The name of the file from which the object will be imported. |
| CONFIRM | Specifies whether or not to display a confirmation dialog before replacing an existing object. If CONFIRM is not specified or is NO, the corresponding resource limit is used. |
| COMMENT | Specifies a comment with the imported object. Enclose the text of the comment in quotes. |
| SHARE | Specifies whether other users are allowed to use the imported object. |
| LANGUAGE | Specifies whether a form is imported in English or in the current session language. A form that is imported in English can be run in any NLF session. A form imported in the session language can only be run in a session of the same language. The default value is provided by the <i>DSQEC_FORM_LANG</i> global variable. |
| ACTION | Specifies whether the entire database table will be replaced or the new data will be appended to the existing table. |
| LOBSFROM | The location to save the LOBs. |
| SPACE | <p>Specifies both the database name and the table space name to save the table in a particular database container and table space.</p> <p>Note: The table space name that you specify must match the default table space name that is set for your user ID on the Save Data tab of the resource limits settings. If you have the permission to override the default table space name, you can specify any table space name to which you want to save the table. The permission to override the table space name is set on the Save Data tab of the resource limits settings.</p> <p>Note:</p> |
| ACCELERATORDATABASE | Specifies the name of the accelerator database that you want to use to save your tables. The ACCELERATORDATABASE keyword can be up to 128 characters long. The default value of the ACCELERATORDATABASE parameter is taken from the <i>DSQEC_SAV_ACCELDB</i> global variable. If the <i>DSQEC_SAV_ACCELDB</i> global variable value is not empty, the database specified by the SPACE keyword is ignored. If the ACCELERATORDATABASE parameter value is not empty, the value of the <i>DSQEC_SAV_ACCELDB</i> is ignored. |

Example

```
IMPORT QUERY FROM C:\Queries\query1.qry
```

LIMIT LOCAL command

The **LIMIT LOCAL** command specifies values for existing local variables.

The parameters set using this command are the only ones that will display in the **Prompt Variables** window when you run the object.

Any new local variables created are valid only for the current query, report, or dashboard.

Syntax

```
>>-- LIMIT LOCAL ( VariableName = Value, ... -----<<
```

| Parameter | Description |
|--------------|--|
| VariableName | The name of the local variable to set available values. |
| Value | The value or list of values which will be available to select from in the Prompt Variables window |

Example

```
LIMIT LOCAL (id=10;20;30
LIMIT LOCAL (TableName=Q.STAFF;Q.INTERVIEW;
```

MAIL TO Command

The MAIL TO command sends the specified object as an Internet Mail attachment.

The specified object can be stored in a database, repository or can be opened in a window (in temporary storage).

Syntax

To send an object that resides in a database:

```
>>-- MAIL ----- ObjectName TO Address -----<<
      +- QUERY +-      ( +- FROM = Address -----+
      +- PROC  +-      +- CCLIST = Address1;Address2 -----+
      +- FORM  +-      +- SUBJECT = Subject -----+
      +-      +-      +- BODY = Text -----+
      +-      +-      +- FORMAT = Text/HTML -----+
      +-      +-      +- SMTPSERVER = SMTP Server -----+
      +-      +-      +- SMTPPORT = SMTP Server port number +-+
      +-      +-      +- SMTPUSER = SMTP Username -----+
      +-      +-      +- SMTPPASSWORD = SMTP Password -----+
      +-      +-      +- DATEFORMAT = Java date format string +-+
      +-      +-      +- TIMEFORMAT = Java time format string +-+
```

```
>>-- MAIL ----- ObjectName TO Address -----<<
      +- VISUAL REPORT +- ( +- FROM = Address -----+
      +-      +-      +- CCLIST = Address1;Address2 -----+
      +-      +-      +- SUBJECT = Subject -----+
      +-      +-      +- BODY = Text -----+
      +-      +-      +- FORMAT = Text/HTML -----+
      +-      +-      +- SMTPSERVER = SMTP Server -----+
      +-      +-      +- SMTPPORT = SMTP Server port number +-+
      +-      +-      +- SMTPUSER = SMTP Username -----+
      +-      +-      +- SMTPPASSWORD = SMTP Password -----+
      +-      +-      +- TYPE = PDF/HTML-----+
      +-      +-      +- METHOD = SPLIT/CONT -----+
```

To send an object that currently resides in an open window (in temporary storage):

```
>>-- MAIL --- QUERY --- TO Address -----<<
      +- PROC +-      ( +- FROM = Address -----+
      +- FORM +-      +- CCLIST = Address1;Address2 -----+
      +-      +-      +- SUBJECT = Subject -----+
      +-      +-      +- BODY = Text -----+
      +-      +-      +- FORMAT = Text/HTML -----+
      +-      +-      +- SMTPSERVER = SMTP Server -----+
      +-      +-      +- SMTPPORT = SMTP Server port number +-+
      +-      +-      +- SMTPUSER = SMTP Username -----+
      +-      +-      +- SMTPPASSWORD = SMTP Password -----+
```



```

+- DATEFORMAT = Java date format string -+
+- TIMEFORMAT = Java time format string -+

```

```

>>-- MAIL --- REPORT --- TO Address -----<<
( +- FROM = Address -----+
  +- CCLIST = Address1;Address2 -----+
  +- SUBJECT = Subject -----+
  +- BODY = Text -----+
  +- FORMAT = Text/HTML -----+
  +- SMTPSERVER = SMTP Server -----+
  +- SMTPPORT = SMTP Server port number ---+
  +- SMTPUSER = SMTP Username -----+
  +- SMTPPASSWORD = SMTP Password -----+
  +- DATEFORMAT = Java date format string -+
  +- TIMEFORMAT = Java time format string -+
  +- TYPE = PDF/HTML-----+
  +- METHOD = SPLIT/CONT -----+

```

```

>>-- MAIL --- VISUAL REPORT --- TO Address -----<<
( +- FROM = Address -----+
  +- CCLIST = Address1;Address2 -----+
  +- SUBJECT = Subject -----+
  +- BODY = Text -----+
  +- FORMAT = Text/HTML -----+
  +- SMTPSERVER = SMTP Server -----+
  +- SMTPPORT = SMTP Server port number -+
  +- SMTPUSER = SMTP Username -----+
  +- SMTPPASSWORD = SMTP Password -----+
  +- TYPE = PDF/HTML-----+
  +- METHOD = SPLIT/CONT -----+

```

```

>>-- MAIL --- DATA --- TO Address -----<<
( +- FROM = Address -----+
  +- CCLIST = Address1;Address2 -----+
  +- DATAFORMAT = CSV/DBF/HTML/IXF/PDF/QMF/SHP/TEXT/WQML/XLS/XLSX/XML --+
  +- SUBJECT = Subject -----+
  +- BODY = Text -----+
  +- FORMAT = Text/HTML -----+
  +- SMTPSERVER = SMTP Server -----+
  +- SMTPPORT = SMTP Server port number -----+
  +- SMTPUSER = SMTP Username -----+
  +- SMTPPASSWORD = SMTP Password -----+
  +- DATEFORMAT = Java date format string -----+
  +- TIMEFORMAT = Java time format string -----+

```

```

>>-- MAIL --- MESSAGE --- TO Address -----<<
( +- FROM = Address -----+
  +- CCLIST = Address1;Address2 -----+
  +- SUBJECT = Subject -----+
  +- BODY = Text -----+
  +- FORMAT = Text/HTML -----+
  +- SMTPSERVER = SMTP Server -----+
  +- SMTPPORT = SMTP Server port number -+
  +- SMTPUSER = SMTP Username -----+
  +- SMTPPASSWORD = SMTP Password -----+
  +- ATTACHMENT = File1;File2 -----+

```

| Parameter | Description |
|------------|--|
| ObjectName | The name of the object to send. |
| FROM | The email of the sender. |
| CCLIST | The email addresses of the recipients. |

| Parameter | Description |
|--------------|--|
| DATAFORMAT | <p>Specifies the file format of the attached data object. If you omit this parameter, the DSQQW_EXP_DT_FRMT global variable supplies the format to be used. You can specify in DSQQW_EXP_DT_FRMT global variable:</p> <ul style="list-style-type: none"> • zero (0) for TEXT format • two (2) for HTML format • three (3) for CSV format • four (4) for IXF format • five (5) for dbase III files • six (6) for Shapefile format • seven (7) for WQML format • eight (8) for XML format • nine (9) for PDF format • ten (10) for XLS format • eleven (11) for XLSX format <p>If you specify IXF, the DSQQW_EXP_OUT_MDE global variable can be set to either zero (0) for character mode System/370 IXF or one (1) for PC/IXF. When exporting reports, HTML format adds the <PRE> tag, TEXT is for a text-based file, and PDF is for a PDF-based file.</p> |
| SUBJECT | The email subject line reference. |
| BODY | The contents of the email message. |
| FORMAT | The email format. Supported formats are Text and HTML. |
| SMTPSERVER | The name of a predefined SMTP server or address of a custom SMTP server. |
| SMTPPORT | The SMTP server port number. |
| SMTPUSER | The user name that is used to authorize on SMTP server. |
| SMTPPASSWORD | The password that is used to authorize on SMTP server. |
| ATTACHMENT | The name and path to the files that will be attached. |
| TYPE | The types that specifies the exported files are visual reports. |
| METHOD | The method separated pages. |
| DATEFORMAT | <p>Specifies how the date is formatted in the HTML, CSV or TXT export file.</p> <p>Date formats are specified by Java date pattern strings. Within date pattern strings, unquoted letters from 'A' to 'Z' and from 'a' to 'z' are interpreted as pattern letters representing the components of a date string.</p> <p>Text can be quoted using single quotes (') to avoid interpretation.</p> <p>Note: If the format string includes spaces, enclose it in quotes. For more information about Java format strings, see the Java 2 SDK, Standard Edition Documentation.</p> |

| Parameter | Description |
|------------|---|
| TIMEFORMAT | <p>Specifies how the time is formatted in the HTML, CSV or TXT export file.</p> <p>Time formats are specified by Java time pattern strings. Within time pattern strings, unquoted letters from 'A' to 'Z' and from 'a' to 'z' are interpreted as pattern letters representing the components of a time string.</p> <p>Text can be quoted using single quotes (') to avoid interpretation.</p> <p>Note: If the format string includes spaces, enclose it in quotes. For more information about Java format strings, see the Java 2 SDK, Standard Edition Documentation.</p> |

Example 1

The following is an example of sending a query, which is opened in the query editor:

```
MAIL QUERY TO abc@mail.com ( SUBJECT="Opened Query"
+ SMTPSERVER=smtp.example.com
```

Example 2

The following is an example of using DATEFORMAT and TIMEFORMAT parameters:

```
MAIL DATA TO abc@mail.com (DATEFORMAT="dd.MM.yy" TIMEFORMAT="hh:mm"
+ BODY="DATA"
+ SUBJECT="DATA"
+ SMTPSERVER=smtp.example.com FROM=cba@mail.com
```

Example 3

The following example sends an image to the specified email address in the HTML format:

```
MAIL MESSAGE TO abc@mail.com (
+ SUBJECT="Image"
+ BODY="<h1>image</h1><br><img src='cid:image.png'></img>"
+ FROM=cba@mail.com
+ SMTPSERVER="smtp.example.com"
+ SMTPUSER=user
+ SMTPPASSWORD=password
+ ATTACHMENT="D:/image.png"
+ FORMAT=HTML
```

PRINT command

The PRINT command prints a copy of an object that is in the current window (temporary storage) or stored in a database.

Syntax

Note: This command is not supported in QMF for WebSphere.

To print a copy of an object that resides on a database:

```
>>-- PRINT ----- ObjectName -----<<
      ( +- PRINTER = Printer -----+
        +- LEFTMARGIN = integer -----+
        +- RIGHTMARGIN = integer -----+
        +- TOPMARGIN = integer -----+
        +- BOTTOMMARGIN = integer -----+
        +- COPIES = NumCopies-----+
        +- WIDTH = NumChars/CONT/AUTO --+
        +- HEIGHT = NumChars/CONT/AUTO --+

>>-- PRINT ----- ObjectName -----<<
      +- QUERY --+      ( +- PRINTER = Printer -----+
```

```

+- PROC +-          +- LEFTMARGIN = integer -----+
+- TABLE +-       +- RIGHTMARGIN = integer -----+
                                     +- TOPMARGIN = integer -----+
                                     +- BOTTOMMARGIN = integer -----+
+

```

```

>>-- PRINT ----- ANALYTICS --- ObjectName -----<<
      ( +- WIDTH = NumChars/CONT/AUTO +-
        +- HEIGHT = NumChars/CONT/AUTO +-
        +- PRINTER = Printer -----+

```

```

>>-- PRINT FORM --- ObjectName-----<<
      ( +- DATETIME = YES/NO -----+
        +- PAGENO = YES/NO -----+
        +- PRINTER = Printer -----+
        +- LENGTH = NumLines/CONT/AUTO -----+
        +- WIDTH = NumChars/CONT/AUTO -----+
        +- LEFTMARGIN = integer -----+
        +- RIGHTMARGIN = integer -----+
        +- TOPMARGIN = integer -----+
        +- BOTTOMMARGIN = integer -----+

```

To print a copy of the current object in an open window (temporary storage):

```

>>-- PRINT ----- QUERY -----<<
      +- PROC +-          ( +- PRINTER = Printer -----+
      +- TABLE +-       +- LEFTMARGIN = integer ---+
                          +- RIGHTMARGIN = integer --+
                          +- TOPMARGIN = integer ----+
                          +- BOTTOMMARGIN = integer -+

```

```

>>-- PRINT REPORT -----<<
      ( +- DATETIME = YES/NO -----+
        +- PAGENO = YES/NO -----+
        +- PRINTER = Printer -----+
        +- COPIES = NumCopies-----+
        +- TYPEFACE = fontname -----+
        +- SIZE = FontSize -----+
        +- BOLD = YES/NO -----+
        +- ITALIC = YES/NO -----+
        +- CHARSET = ANSI/DEFAULTS/SYMBOL/SHIFTJIS/GB2312/ +-
                    HANGUEL/CHINESEBIG5/OEM/JOHAB/HEBREW/ --+
                    ARABIC/GREEK/TURKISH/THAI/EASTEUROPE/ --+
                    RUSSIAN/MAC/BALTIC -----+
        +- ORIENTATION = PORTRAIT/LANDSCAPE -----+
        +- USEFORMPS = YES/NO -----+
        +- LENGTH = NumLines/CONT/AUTO -----+
        +- WIDTH = NumChars/CONT/AUTO -----+
        +- LEFTMARGIN = integer -----+
        +- RIGHTMARGIN = integer -----+
        +- TOPMARGIN = integer -----+
        +- BOTTOMMARGIN = integer -----+

```

```

>>-- PRINT FORM -----<<
      ( +- DATETIME = YES/NO -----+
        +- PAGENO = YES/NO -----+
        +- PRINTER = Printer -----+
        +- LENGTH = NumLines/CONT/AUTO -----+
        +- WIDTH = NumChars/CONT/AUTO -----+
        +- LEFTMARGIN = integer -----+
        +- RIGHTMARGIN = integer -----+
        +- TOPMARGIN = integer -----+
        +- BOTTOMMARGIN = integer -----+

```

| Parameter | Description |
|------------|----------------------------------|
| ObjectName | The name of the object to print. |

| Parameter | Description |
|--------------|---|
| DATETIME | Specifies whether or not to include the current date and time in the page footer. When you print a report, the date and time are printed in the page footer unless you specify DATETIME=NO or include the &DATE or &TIME variables in the page text in the form. If included, the date and time are formatted according to the current Windows regional settings. |
| PAGENO | Specifies whether or not to include page numbers in the page footer. When you print a report, page numbers are printed in the page footer unless you specify PAGENO=NO or include the &PAGE variable in the page text in the form. |
| PRINTER | Specifies the name of the printer on which to print the object or report. Tip: You can get the name of the printer from the General tab of the Print window. Put the name of your printer in double quotation marks if you want to specify it as the value of the PRINTER parameter. For example, PRINTER="My_Printer" |
| TYPEFACE | Specifies the font typeface to use when printing the object. |
| COPIES | Specifies the number of copies to print when printing the object. |
| SIZE | Specifies the font size to use when printing the object. |
| BOLD | Specifies to use bold text when printing the object. |
| ITALIC | Specifies to use italic text when printing the object. |
| CHARSET | Specifies the character set to use when printing the object. |
| ORIENTATION | Specifies the paper orientation of the printed object. |
| USEFORMPS | Specifies whether or not to use the Windows form associated with the object. |
| LENGTH | Specifies the maximum number of lines to print on each page (from 1 to 999), CONT (to cause continuous printing without page breaks), or AUTO. If LENGTH is omitted, the values of the <i>DSQW_RPT_LEN_TYP</i> and <i>DSQW_RPT_NUM_LNS</i> global variables are used. |
| WIDTH | <ul style="list-style-type: none"> Specifies the maximum number of characters to print on each line (from 1 to 999), CONT (to cause continuous printing without page breaks), or AUTO. When you print a report, lines longer than WIDTH are formatted on a subsequent page. <i>For Analytics object:</i> Specifies the width of a printed page. The unit of width is one single-byte character. |
| HEIGHT | <i>For Analytics object:</i> Specifies the height of a printed page. The unit of height is one single-byte character. |
| LEFTMARGIN | Specifies the value that will be used for the left margin. Units of measurement are based on inches. |
| RIGHTMARGIN | Specifies the value that will be used for the right margin. Units of measurement are based on inches. |
| TOPMARGIN | Specifies the value that will be used for the top margin. Units of measurement are based on inches. |
| BOTTOMMARGIN | Specifies the value that will be used for the bottom margin. Units of measurement are based on inches. |

Example

The following examples print a copy of the current objects in an open window. If you want to print an object from the database, specify its name in the procedure.

```
PRINT QUERY
PRINT REPORT (LENGTH=60, DATETIME=NO)
```

RESET GLOBAL

The RESET GLOBAL command deletes the names and values of user global variables.

Syntax

To reset specific global variables:

```
>>-- RESET GLOBAL (VarName1, VarName2, ... -----<<
```

To reset all global variables:

```
>>-- RESET GLOBAL ALL -----<<
```

| Parameter | Description |
|--------------|---|
| VariableName | The names of the specific variables that will be deleted. You can name up to 10 variables previously set by the SET GLOBAL command. ALL deletes the names and values of all the variables previously set by the SET GLOBAL command. Variables whose names begin with DSQ are restricted, and can not be deleted. |

Example

```
RESET GLOBAL (Variable1, Variable2
```

```
RESET GLOBAL ALL
```

RUN command

The RUN command runs procedures, queries, or dynamarts that are stored in a database or are current objects in an open window (reside in temporary storage). For queries, the RUN command provides the ability to save the retrieved results to a database.

Syntax

To run an object that resides on the database:

```
>>-- RUN ----- ObjectName -----<<
      +- QUERY +-      ( +- &&Variable = Value -----+
                        +- ACCELERATOR = acceleratorname -----+
                        +- ACCELERATORDATABASE = databasename -----+
                        +- ACTION = REPLACE/APPEND -----+
                        +- COMMENT = text -----+
                        +- CONFIRM = YES/NO -----+
                        +- FORM = FORM/formname-----+
                        +- METHOD = REGULAR/FAST/FASTSAFE-----+
                        +- MODE = GRID/RAW -----+
                        +- ROWIDADD = YES/NO -----+
                        +- ROWIDDISP = EXCLUDE/CONVERT/ALWAYS/BYDEFAULT-----+
                        +- ROWIDNAME = text -----+
                        +- ROWLIMIT = NumRows-----+
                        +- SCOPE = NumChars -----+
                        +- SPACE = tablespaces/database.tablespace-----+
```

```

+-- SPACE DATABASE = database-----+
+-- TABLE = name/owner.name-----+

>>-- RUN ----- ObjectName -----<<
+-- PROC +- ( + &&Variable = Value -----+

>>-- RUN ----- ObjectName -----<<
+-- DYNAMART +- ( +- &&Variable = Value -----+
+-- CONFIRM = YES/NO -----+
+-- REFRESH = YES/NO -----+
+-- ROWLIMIT = NumRows -----+

```

To run an object from an open window (temporary storage):

```

>>-- RUN QUERY -----<<
( +- &&Variable = Value -----+
+-- ACCELERATOR = acceleratorname -----+
+-- ACCELERATORDATABASE = databasename -----+
+-- ACTION = REPLACE/APPEND -----+
+-- COMMENT = text -----+
+-- CONFIRM = YES/NO -----+
+-- FORM = FORM/formname -----+
+-- METHOD = REGULAR/FAST/FASTSAFE-----+
+-- MODE = GRID/RAW -----+
+-- ROWIDADD = YES/NO -----+
+-- ROWIDDISP = EXCLUDE/CONVERT/ALWAYS/BYDEFAULT-----+
+-- ROWIDNAME = text -----+
+-- ROWLIMIT = NumRows/NO -----+
+-- SCOPE = NumChars -----+
+-- SPACE = tablespace/database.tablespace-----+
+-- SPACE DATABASE = database-----+
+-- TABLE = name/owner.name-----+

>>-- RUN PROC -----<<
( +- &&Variable = Value ... -----+

>>-- RUN DYNAMART -----<<
( +- &&Variable = Value -----+
+-- CONFIRM = YES/NO -----+
+-- REFRESH = YES/NO -----+
+-- ROWLIMIT = NumRows -----+

```

Table 17. Parameters for running objects

| Parameter | Description |
|------------|---|
| &&Variable | Assigns a value to a variable in the query, procedure, or dynamart that is run. The variable name can be from 1 to 17 characters long and the value can be from 1 to 55 characters long. You can specify any number of variables and values on the RUN command. If there are variables in the query, procedure, or dynamart that are not given values on the RUN command, and are not global variables, the user will be prompted for the values. When you include variable assignments in a procedure, you must use two ampersands to prevent variable substitution before the procedure is run. |
| CONFIRM | Specifies whether or not to display a confirmation dialog before replacing or changing an object as a result of this command. If CONFIRM is not specified or is NO, the corresponding resource limit is used. |
| FORM | Specifies a form to use when generating a displayed report using the data retrieved by the query. You can specify the keyword FORM to use the current form object, or specify the name of a form saved in the database. |
| ObjectName | The name of the query, procedure, or dynamart to run. |
| REFRESH | Specifies whether or not to refresh data previously saved in a dynamart. If this parameter is set to YES, the dynamart is opened in the Query editor, query is run to refresh data, and then updated data is automatically saved in the dynamart. |

Table 17. Parameters for running objects (continued)

| Parameter | Description |
|-----------|---|
| ROWLIMIT | Specifies the maximum number of rows to retrieve for the query or dynamart. |

Table 18. Parameters for saving query results to a database

| Parameter | Description |
|---------------------|---|
| ACCELERATORDATABASE | Specifies the name of the accelerator database that you want to use to save your tables. The ACCELERATORDATABASE keyword can be up to 128 characters long. The default value of the ACCELERATORDATABASE parameter is taken from the DSQEC_SAV_ACCELDB global variable. If the DSQEC_SAV_ACCELDB global variable value is not empty, the database specified by the SPACE keyword is ignored. If the ACCELERATORDATABASE parameter value is not empty, the value of the DSQEC_SAV_ACCELDB is ignored. |
| ACTION | Specifies whether to replace the entire database table or append data to the existing table. <ul style="list-style-type: none"> Specify REPLACE to replace the data in the database table by the query results data. Specify APPEND to add query results to the existing database table. |
| COMMENT | Specifies a comment for the database table to which you save query results. The text of the comment must be enclosed in quotes. |
| CONFIRM | Specifies whether to display a confirmation dialog before replacing or changing an object as a result of this command. If CONFIRM is not specified or is NO, the corresponding resource limit is used. |
| METHOD | Specifies the method of saving the query results data. <ul style="list-style-type: none"> Specify REGULAR to send query results data from the client back to the database server and insert them into the table. Specify FAST to rerun the query at the server and insert query results directly into the table. Specify FASTSAFE to rerun the query at the server without the ORDER BY clauses and insert query results directly into the table. |
| MODE | Specifies whether the query result data is saved with formatting and added calculated columns. <ul style="list-style-type: none"> Specify GRID to set that all of the data as it is currently formatted in the current query results will be saved. Any calculated columns that have been added to the query results are included. Specify RAW to set that all of the data in the current query results will be saved. Any formatting that is applied to the data will not be saved. Any calculated columns that are added to the query results will not be saved. This is the default value. |
| ROWIDADD | Specifies whether to add the Row ID column to the table. |
| ROWIDNAME | Specifies the name for the new Row ID column. |
| ROWIDDISP | Specifies the disposition of the new Row ID column. |
| SCOPE | Specifies the commit scope of the data. |

Table 18. Parameters for saving query results to a database (continued)

| Parameter | Description |
|-----------|---|
| SPACE | <p>Specifies both the database name and the table space name to save the table in a particular database container and table space.</p> <p>Note: The table space name that you specify must match the default table space name that is set for your user ID on the Save Data tab of the resource limits settings. If you have the permission to override the default table space name, you can specify any table space name to which you want to save the table. The permission to override the table space name is set on the Save Data tab of the resource limits settings.</p> <p>Note:</p> |
| TABLE | <p>Specifies the name of the database table to which you want to save query results. If you specify both owner name and table name as a value for this parameter, the table is saved with the specified name for the specified owner. If you specify only the table name, the table is save with the specified name for the owner that runs the procedure.</p> <p>Note: If you specify the TABLE parameter for your query, the ROWLIMIT parameter is ignored.</p> |

Example

The following example runs a query from the QMF Catalog.

```
RUN QUERY USER1.QUERY1 (&&Var1=10
```

RUNTSO command

You issue the **RUNTSO** command to start the Q.DSQQMFSP stored procedure from a CALL statement You use the **RUNTSO** command to pass the name of a query or procedure to run on QMF for TSO. The query or procedure that is named in this command must exist in the QMF catalog in the subsystem against which the RUNTSO command runs.

RUNTSO, returns one result set each time the following command is encountered in the procedure:

```
PRINT REPORT (PRINTER=' ')
```

Classic forms are created on the basis of the returned result sets. These classic forms are generated without any additional form formats, but they can be specified in the **Form design** tab.

Note: Support for this command requires that the stored procedure interface to QMF for TSO be installed. Additionally, if your QMF for TSO installation does not use the default command synonyms table, Q.COMMAND_SYNONYMS, any synonym you issue must have a definition stored in the table that is named in the **SYNONYMS** field of the user profile. If the synonym definition is not found in this table, the command will fail. For information about installing the stored procedure interface, see the chapter “Installing the QMF stored procedure interface (TSO only)” in *Installing and Managing Db2 QMF for TSO and CICS (GC19-2886-04)*.

Syntax

Use the following syntax to include the **RUNTSO** command in a procedure.

```
>>-RUNTSO--ObjectName--(-+-----+----->
| .-=-. .-NONE-. |
| -TRACELEVEL-+-----+L2-+-- |
```


| Parameter | Description |
|-----------------|---|
| LANGUAGE | <p>Specifies the language in which QMF will run.</p> <p>This parameter can be a one-letter language identifier from the following table.</p> <p>The default value depends on the <i>DSQEC_NLFCMD_LANG</i> variable. For example, if <i>DSQEC_NLFCMD_LANG</i>=0 then <i>DSQAO_NLF_LANG</i> is used as the language.</p> <p>If <i>DSQEC_NLFCMD_LANG</i>=1 then E is used.</p> |

The following table lists the supported languages supported and the associated language identifier.

Table 19. Language-to-language identifier

| National language feature | Name that QMF uses for this NLF | Language ID |
|---------------------------|---------------------------------|-------------|
| English | ENGLISH | E |
| U/C English | UPPERCASE | U |
| Danish | DANSK | Q |
| Canadian French | FRANCAIS CANADIEN | C |
| French | FRANCAIS | F |
| German | DEUTSCH | D |
| Italian | ITALIANO | I |
| Japanese | NIHONGO | K |
| Korean | HANGEUL | H |
| Brazilian Portuguese | PORTUGUES | P |
| Spanish | ESPAÑOL | S |
| Swedish | SVENSKA | V |
| Swiss French | FRANCAIS (SUISSE) | Y |
| Swiss German | DEUTSCH (SCHWEIZ) | Z |

Examples

The following example is of the **RUNTSO** command used to pass Q .STAFF query to run on QMF for TSO

```
RUNTSO Q.STAFF (TRACELEVEL=NONE L2DESTINATION=NONE LANGUAGE=E
```

The following examples show how to use the object name that contains parameters in the **RUNTSO** command:

Note: The STAFFPROC is the example QMF procedure that runs the Q .STAFF query.

- The following example passes the parentheses as delimiters for the variable values:

```
RUNTSO "STAFFPROC(&NAME=('PERNAL'),&DEPT=(20),&JOB=('SALES'))" (LANGUAGE=E TRACELEVEL=L2 L2DESTINATION=NONE
```

The example command returns a result set containing the following row from the Q.STAFF sample table:

| ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
|----|--------|------|-------|-------|----------|--------|
| 20 | PERNAL | 20 | SALES | 8 | 18171.25 | 612.45 |

- The following example passes variable values without using parentheses as delimiters:

```
RUNTSO "STAFFPROC(&NAME='PERNAL',&DEPT=20,&JOB='SALES')" (LANGUAGE=E TRACELEVEL=L2 L2DESTINATION=NONE
```

- The following example passes a variable value that contains an apostrophe:

```
RUNTSO "STAFFPROC(&NAME='O' 'BRIEN',&DEPT=38,&JOB='SALES')" (LANGUAGE=E TRACELEVEL=L2 L2DESTINATION=NONE
```

This command returns the following row as the first result set:

| ID | NAME | DEPT | JOB | YEARS | SALARY | COMM |
|----|---------|------|-------|-------|----------|--------|
| 40 | O'BRIEN | 38 | SALES | 6 | 18006.00 | 846.55 |

SAVE command

The SAVE command saves an object from temporary storage to a database.

Syntax

```
>>-- SAVE --- QUERY --- AS ObjectName -----<<
      +- PROC +-                               ( +- COMMENT = text -----+
                                               +- CONFIRM = YES/NO -----+
                                               +- SHARE = YES/NO -----+
                                               +- FOLDER = foldername +-
```

```
>>-- SAVE -- FORM -- AS ObjectName -----<<
      ( +- COMMENT = text -----+
        +- CONFIRM = YES/NO -----+
        +- SHARE = YES/NO -----+
        +- LANGUAGE = ENGLISH/SESSION +-
        +- FOLDER = foldername -----+
```

```
>>-- SAVE -- DATA -- AS TableName -----<<
      ( +- COMMENT = text -----+
        +- CONFIRM = YES/NO -----+
        +- METHOD = REGULAR/FAST/FASTSAFE-----+
        +- SCOPE = NumChars -----+
        +- RESULTSET = integer -----+
        +- ROWIDADD = YES/NO -----+
        +- ROWIDNAME = text -----+
        +- ACCELERATOR = acceleratorname -----+
        +- ACCELERATORDATABASE = databasename -----+
        +- ROWIDDISP = EXCLUDE/CONVERT/ALWAYS/BYDEFAULT-----+
        +- ACTION = REPLACE/APPEND -----+
        +- SPACE = tablespace/database.tablespace-----+
        +- SPACE DATABASE = database-----+
        +- MODE = GRID/RAW -----+
```

To save visual reports created in QMF for Windows:

```
>>-- SAVE -- REPORT -- AS ObjectName -----<<
      ( +- COMMENT = text -----+
        +- CONFIRM = YES/NO -----+
```

| Parameter | Description |
|------------|---|
| ObjectName | The name to assign to the object (query, form, procedure, or table) when it is saved. |

| Parameter | Description |
|-----------|--|
| COMMENT | Specifies a comment with the saved object. Enclose the text in quotes. |
| CONFIRM | Specifies whether or not to display a confirmation dialog before replacing or changing an object as a result of this command. If CONFIRM is not specified or is NO, the corresponding resource limit is used. |
| SHARE | Specifies whether other users are allowed to use the saved object. |
| LANGUAGE | Specifies whether a form is saved in English or in the current session language. A form that is saved in English can be run in any NLF session. A form that is saved in the session language can only be run in a session of the same language. The default value is provided by the DSQEC_FORM_LANG global variable. |
| FOLDER | Specifies the folder in QMF Catalog to that links to the objects must be saved. The default value is provided by the DSQEC_CURR_FOLDER global variable. |
| METHOD | Specifies the method of saving the data. <ul style="list-style-type: none"> • REGULAR sends data from the client back to the database server to be inserted into the table. • FAST reruns the query at the server and inserts the results directly into the table. • FASTSAFE reruns the query at the server without the ORDER BY clauses and inserts the results directly into the table. |
| SCOPE | Specifies the commit scope of the data. |
| RESULTSET | Specifies the number of the result set to save in queries that have multiple result sets. If you omit this parameter, the current result set will be saved. |
| ROWIDADD | Specifies whether to add the Row ID column to the table. |
| ROWIDNAME | Specifies the name to assign to the new Row ID column. |
| ROWIDDISP | Specifies the disposition of the new Row ID column. |
| ACTION | Specifies whether to replace the entire database table or append data to the existing table. |
| SPACE | Specifies both the database name and the table space name to save the table in a particular database container and table space. <p>Note: The table space name that you specify must match the default table space name that is set for your user ID on the Save Data tab of the resource limits settings. If you have the permission to override the default table space name, you can specify any table space name to which you want to save the table. The permission to override the table space name is set on the Save Data tab of the resource limits settings.</p> <p>Note:</p> |

| Parameter | Description |
|---------------------|---|
| MODE | <p>Specifies whether the query result data is saved with formatting and added calculated columns. You can specify one of the following values:</p> <ul style="list-style-type: none"> • GRID to specify that all of the data as it is currently formatted in the current query results will be saved. Any calculated columns that have been added to the query results are included. • RAW to specify that all of the data in the current query results will be saved. Any formatting that has been applied to the data will not be saved. Any calculated columns that have been added to the query results will not be saved. This is the default value. |
| ACCELERATORDATABASE | <p>Specifies the name of the accelerator database that you want to use to save your tables. The ACCELERATORDATABASE keyword can be up to 128 characters long. The default value of the ACCELERATORDATABASE parameter is taken from the DSQEC_SAV_ACCELDB global variable. If the DSQEC_SAV_ACCELDB global variable value is not empty, the database specified by the SPACE keyword is ignored. If the ACCELERATORDATABASE parameter value is not empty, the value of the DSQEC_SAV_ACCELDB is ignored.</p> |

Example

```
SAVE QUERY AS USER1.QUERY1 (CONFIRM=NO
```

SET GLOBAL command

The SET GLOBAL command sets the values of existing global variables or creates variables and values.

Syntax

```
>>-- SET GLOBAL ( VariableName = Value, ... -----<<
+- LIFETIME = CURRENT/PERMANENT -+
```

| Parameter | Description |
|--------------|--|
| VariableName | The name of the global variable to set or create. Global variable names can be from 1 to 17 characters long. Variables whose names begin with DSQ are restricted, and can not be created or deleted. |
| Value | The value to give the global variable. The value can be from 1 to 55 characters long. The values of variables whose names begin with DSQ are restricted. See the "Working with global variables" topic for more information. |
| LIFETIME | <p>The period during which the variable is available for use:</p> <ul style="list-style-type: none"> • CURRENT - the variable is available in the current session (default value). • PERMANENT - the variable is available persistently. |

Example

```
SET GLOBAL (Var1=abc, Var2=def LIFETIME = PERMANENT
```

SET INVISIBLE command

The **SET INVISIBLE** command hides specified variables from the **Prompt Variables** dialog.

The **Prompt Variables** dialog will not request any value for the specified variables. If all local variables are set invisible, the **Prompt Variables** dialog does not display.

For example, if you have a query `select * from q.staff`, where `id=&id` or `id=&visible` and if you run a procedure `run query (&&id=20)`, then you will be prompted to enter value for `id` and `visible` variables. Also the `id` variable will be defaulted to `20`. But if you run the following procedure: `set invisible (id run query (&&id=20)`, then you will be prompted to enter a value for the `visible` variable only. The `id` variable will be set to `20` and will be hidden from the dialog. Also, if you run the following procedure: `set invisible (id visible run query (&&id=20)`, then you will not be prompted to enter any variable values. The `id` variable will be set to `20` and the `visible` variable will be set to an empty string.

Syntax

```
>>-- SET INVISIBLE ( VariableName1, VariableName2, ... -----<<
```

Parameters

| Parameter | Description |
|--------------|---|
| VariableName | The name of variable not to be displayed in the Prompt Variables dialog. |

Example

```
SET INVISIBLE (Var1, Var2
```

SET LOCAL command

The **SET LOCAL** command sets the values of existing local variables or creates new variables and values.

Variables specified in **SET LOCAL** are available only for the current object (query, report, or dashboard) and do not appear in Global Variables list or affect other procedures.

Syntax

```
>>-- SET LOCAL ( VariableName = Value, ... -----<<
```

Parameters

| Parameter | Description |
|--------------|--|
| VariableName | The name of the local variable to set or create. Local variable names can be from 1 to 17 characters long. Variables whose names begin with DSQ are restricted, and can not be created or deleted. |
| Value | The value to give the local variable. The value can be from 1 to 55 characters long. The values of variables whose names begin with DSQ are restricted. |

Example

```
SET LOCAL (Var1=abc, Var2=def
```

SET LOCAL WITH VALUES command

The **SET LOCAL WITH VALUES** sets the values that can be specified for local variables.

SET LOCAL WITH VALUES works like LIMIT LOCAL but the **Prompt Variables** dialog allows you select one of specified values or enter another value.

Syntax

```
>>-- SET LOCAL WITH VALUES ( VariableName = Value, ... -----<<
```

Parameters

| Parameter | Description |
|--------------|--|
| VariableName | The name of the local variable to set or create. Local variable names can be from 1 to 17 characters long. Variables whose names begin with DSQ are restricted, and can not be created or deleted. |
| Value | The value or list of values to give the local variables. The value(s) specified will be available to choose from in the Prompt Variables window. The value can be from 1 to 55 characters long. The values of variables whose names begin with DSQ are restricted. |

Example

```
SET LOCAL WITH VALUES (Var1=abc
```

SET OPTIONS command

The SET OPTIONS command specifies procedure execution options.

Syntax

```
>>-- SET OPTIONS -----<<  
      ( +- STOPONERROR = YES|NO -----+  
        +- SUPPRESSMESSAGES=ALL|INFORM|ERROR -+)
```


Parameters

| Parameter | Description |
|------------------|--|
| STOPONERROR | <p>Specifies whether the procedure stops running when an error occurs. Valid values are as follows:</p> <ul style="list-style-type: none"> • YES The procedure stops running when an error occurs. • NO The procedure does not stop running when an error occurs. <p>The processing associated with this parameter is similar to the processing associated with DSQQW_PROC_FAIL_ON_ERROR. However, the value set for STOPONERROR takes precedent over the values set for DSQQW_PROC_FAIL_ON_ERROR.</p> <p>If you do not specify a value for STOPONERROR, the value for DSQQW_PROC_FAIL_ON_ERROR determines whether processing stops or continues when an error occurs.</p> <p>If you do specify a value for STOPONERROR, the value for DSQQW_PROC_FAIL_ON_ERROR is ignored.</p> |
| SUPPRESSMESSAGES | <p>Specifies the types of messages to suppress while the procedure is running. Valid values are as follows:</p> <ul style="list-style-type: none"> • ALL Suppresses all message dialogs when the procedure is running. • INFORM Suppresses information (confirmation) message dialogs when the procedure is running. • ERROR Suppresses error message dialogs when the procedure is running. |

Example

```
SET OPTIONS (STOPONERROR=YES SUPPRESSMESSAGES=ALL
```

SHOW command

The SHOW command displays objects from temporary storage and is similar to the [DISPLAY](#) command.

The SHOW QUERY, SHOW FORM, and SHOW PROC commands activate the window containing the current query, form, or procedure, respectively. The SHOW REPORT command is a synonym for the SHOW FORM command. The SHOW GLOBALS command opens the **Global Variables** dialog.

Syntax

```
>>-- SHOW --- PROC -----<<
      +- GLOBALS +-
      +- FORM ----+
            +- .MAIN-----+
            +- .BREAK1-----+
            +- .BREAK2-----+
            +- .BREAK3-----+
```

```

+- .BREAK4-----+
+- .BREAK5-----+
+- .BREAK6-----+
+- .CALC-----+
+- .COLUMNS-----+
+- .CONDITIONS-+
+- .DETAIL-----+
+- .FINAL-----+
+- .OPTIONS-----+
' - .PAGE-----'

```

```

>>-- SHOW QUERY
-----<<
      ( +- VIEW = BUILD/CHART/DESIGN/DISPLAY/LAYOUT/PROMPTED/RESULTS/SQL/STRUCTURE ---
+
      +- RESULTSET = integer
-----+

```

```

>>-- SHOW REPORT -----<<
      ( VIEW = CLASSIC/VISUAL/QUICK

```

| Parameter | Description |
|-----------|--|
| VIEW | Specifies the appearance of the target query or report. |
| RESULTSET | Specifies the number of the result set that is to be displayed. This parameter is only available if VIEW is equal to RESULTS and the query is a stored procedure call. |

Example

```
SHOW QUERY (VIEW=RESULTS)
```

USE REPOSITORY command

The USE REPOSITORY command establishes a connection to the specified repository.

As a result of running this command, the following actions occur:

- All opened objects (including procedures) are closed
- The current repository connection is stopped
- A new connection to specified repository is established

You can omit the USER/PASSWORD of the repository if it is not secured. You can omit the DBUSER/DBPASSWORD if they are specified (and remembered) for the specified repository connection.

Syntax

```

>>-- USE REPOSITORY ----- RepositoryName -----<<
      (+- USER = ... -----+
      +- PASSWORD = ... -----+
      +- DBUSER = ... -----+
      +- DBPASSWORD = ... -----+

```

Parameters

RepositoryName

The name of repository to connect to. If the name includes spaces, enclose it quotes.

USER

The user name used to connect to the secured repository.

Note: This parameter is required only if the repository is secured.

PASSWORD

The password used to connect to the secured repository.

Note: This parameter is required only if the repository is secured.

DBUSER

The user name used to connect to the specified repository's database.

If you checked **Remember user name and password** in the Repository Connection preferences, you can omit specifying the value here.

Note: The parameter value entered here overrides the value set in the Repository Connection preferences.

DBPASSWORD

The password used to connect to the specified repository's database.

If you checked **Remember user name and password** in the Repository Connection preferences, you can omit specifying the value here.

Note: The parameter value entered here overrides the value set in the Repository Connection preferences.

Example

```
USE REPOSITORY Default
USE REPOSITORY SomeRepository (USER=User PASSWORD=password)
```

JavaScript functions in procedures

The following topics describe the functions for the `proc` object and the result set of the current open query.

You can use these functions to execute procedures, get values of global variables, check tables for the existence, and show prompts or message boxes. You can also use them to work with the result set of the query opened in the query editor to get the number of columns or rows, cell values, column types and names.

Descriptions include syntax diagrams and usage examples:

Related tasks

[Creating a new procedure](#)

You can use the **Procedure** window to create new procedures.

[Creating JavaScript procedures](#)

You can use JavaScript syntax to create more complex and not linear procedures.

[Creating procedures as flowcharts](#)

You can use the enhanced procedure editor to create linear or complex JavaScript procedures as flowcharts.

Related reference

[Procedure commands](#)

In this section, each procedure command is described. Descriptions include syntax diagrams and usage examples.

The `proc.exec()` function

Use the `proc.exec()` function to execute a procedure command.

Syntax

```
proc.exec('procedure_command');
```

| Parameter | Description |
|-------------------|---|
| procedure_command | It specifies the procedure command that you want to execute. For more information about available procedure commands, see the Procedure commands section. |

Example

```
/*JavaScript*/
proc.exec('RUN QUERY "qmf:/.workspaces/Business Analyst View/Visual Queries/EmployeeQ"');
```

The procedure example runs the Q.STAFF query that is saved to Business Analyst View workspace in the Samples repository.

The proc.exists() function

Use the `proc.exists()` function to verify the specified path to the object in the repository, an object from the QMF catalog, or the file for the existence.

Syntax

```
proc.exists('key');
```

| Parameter | Description |
|-----------|---|
| key | It specifies the repository key, an object from the QMF catalog, or the file that is checked for the existence. |

Example

```
/*JavaScript*/
var tableName = proc.prompt('What table do you want check for the existence?', 'Q.STAFF');
if (proc.exists(tableName))
{
    proc.messageBox('The ' + tableName + ' table exists', MB_OK, ID_OK);
}
else
{
    proc.messageBox('The ' + tableName + ' table does not exist', MB_OK, ID_OK);
}
```

The example procedure checks the Q.STAFF table for the existence and displays a message with the result.

The proc.getVariable() function

Use the `proc.getVariable()` function to get the value of the specified variable.

Syntax

```
proc.getVariable('global_variable');
```

| Parameter | Description |
|-----------------|---|
| global_variable | It specifies the global variable the value of which you want to fetch. For more information about global variables, see the <i>System defined global variables</i> section. |

Example

```
/*JavaScript*/
var fetchedRowsNum = proc.getVariable('DSQAO_NUM_FETCHED');
proc.messageBox('The value of DSQAO_NUM_FETCHED is ' + fetchedRowsNum);
```

The value of the DSQAO_NUM_FETCHED global variable is saved to the fetchedRowsNum variable and displayed in the message box.

The proc.getCellValue() function

Use the proc.getCellValue() function to get the value of a cell, which is specified by its row and column in the current result set.

Syntax

```
proc.getCellValue(row, column);
```

| Parameter | Description |
|-----------|--|
| row | It specifies the row of the cell, the value of which you want to get. |
| column | It specifies the column of the cell, the value of which you want to get. |

Example

```
/*JavaScript*/
proc.exec('DISPLAY Q.STAFF');
var row = 30;
proc.messageBox(proc.getCellValue(row, 2) + ' was hired ' + proc.getCellValue(row,5) + ' years ago.');
```

The example procedure displays the Q.STAFF table, finds the values of the columns 2 and 5 in the 30 row, and shows the message: “DAVIS was hired 5 years ago.”

The proc.getColumnName() function

Use the proc.getColumnName() function to get the column name in the current result set.

Syntax

```
proc.getColumnName(column);
```

| Parameter | Description |
|-----------|---|
| column | It specifies the column, the name of which you want to get. |

Example

```
/*JavaScript*/
proc.exec('DISPLAY Q.STAFF');
proc.messageBox('The column with the index 3 is "' + proc.getColumnName(3) + '"', MB_OK, ID_OK);
```

The example procedure displays the Q.STAFF table, finds the column 3, and shows the following message box with the OK button that has focus: 'The column with the index 3 is “DEPT”'.

The `proc.getColumnType()` function

Use the `proc.getColumnType()` function to get the column type as a string in the current result set.

Syntax

```
proc.getColumnType(column);
```

| Parameter | Description |
|-----------|---|
| column | It specifies the column, the type of which you want to get as a string. |

Example

```
/*JavaScript*/  
var columnIndex = 6;  
proc.exec('DISPLAY Q.STAFF');  
proc.messageBox('The column with the index ' + columnIndex +  
' has the database type "' + proc.getColumnType(columnIndex) + '"', MB_OK, ID_OK);
```

The example procedure displays the `Q.STAFF` table, finds the column 6, and shows the following message box with the OK button that has focus: "The column with the index 6 has the database type "DECIMAL (7,2)".

The `proc.getNumColumns()` function

Use the `proc.getNumColumns()` function to get the number of columns of the current result set.

Syntax

```
proc.getNumColumns();
```

Example

```
/*JavaScript*/  
proc.exec('DISPLAY Q.STAFF');  
proc.messageBox('The Q.STAFF table has ' + proc.getNumColumns() + ' columns');
```

The example procedure displays the `Q.STAFF` table and shows the message: "The `Q.STAFF` table has 7 columns."

The `proc.getNumRows()` function

Use the `proc.getNumRows()` function to get the number of rows of the current result set.

Syntax

```
proc.getNumRows();
```

Example

```
/*JavaScript*/  
proc.exec('DISPLAY Q.STAFF');  
proc.messageBox('The Q.STAFF table has ' + proc.getNumRows() + ' rows');
```

The example procedure displays the `Q.STAFF` table and shows the message: "The `Q.STAFF` table has 35 rows."

The `proc.include()` function

Use the `proc.include()` function to add external code to your procedure script.

Syntax

Note: You can use external functions and objects only after you connect the required external JavaScript module with your procedure script with the `proc.include()` function.

```
proc.include(strLocation);
```

| Parameter | Description |
|--------------------------|---|
| <code>strLocation</code> | Specifies either a path to a JavaScript Module in a repository or a link to an external source from the Internet. |

Examples

Example 1. This example includes an external module from the Internet that contains the `algorithm()` function:

```
/*JavaScript*/  
proc.include('http://example.com/algorithm.js');
```

Example 2. This example includes the `algorithm` JavaScript Module from the `qmf:/.workspaces/Business Analyst View/algorithm` repository that contains the `algorithm()` function:

```
/*JavaScript*/  
proc.include("qmf:/.workspaces/Business Analyst View/algorithm");
```

The `proc.messageBox()` function

Use the `proc.messageBox()` function to display a message of the specified type and with default value. This function also returns the identifier of the clicked button.

Syntax

```
proc.messageBox('message', type, defaultValue);
```

| Parameter | Description |
|----------------------|---------------------------------------|
| <code>message</code> | It specifies the text of the message. |

| Parameter | Description |
|--------------|---|
| type | <p>It specifies the combination of the available constants:</p> <ul style="list-style-type: none"> • MB_OK displays a message box with one OK button. • MB_OKCANCEL displays a message box with the OK and Cancel buttons. • MB_ABORTRETRYIGNORE displays a message box with Abort, Retry, and Ignore buttons. • MB_YESNOCANCEL displays a message box with the Yes, No, and Cancel buttons. • MB_YESNO displays a message box with the Yes and No buttons. • MB_RETRYCANCEL displays a message box with the Retry and Cancel buttons. • MB_OKABORTIGNORE displays a message box with the OK, Abort, and Ignore buttons. • MB_COMMITOKCANCEL displays a message box with the Commit, OK, and Cancel buttons. • MB_COPY displays a message box with the Copy button. • MB_ASYNC displays a message box asynchronously with the performed procedure. • MB_ERROR displays an error message box. • MB_WARNING displays a warning message box. |
| defaultValue | <p>It specifies the identifier of the button that has focus by default. The available identifiers are:</p> <ul style="list-style-type: none"> • ID_OK indicates that a user clicked the OK button. • ID_CANCEL indicates that a user clicked the Cancel button. • ID_YES indicates that a user clicked the Yes button. • ID_NO indicates that a user clicked the No button. • ID_ABORT indicates that a user clicked the Abort button. • ID_RETRY indicates that a user clicked the Retry button. • ID_IGNORE indicates that a user clicked the Ignore button. |

Example

```
/*JavaScript*/
proc.messageBox('Message box from the JavaScript procedure.');
```

This example displays a message box with the specified text and one OK button, which has focus.

The proc.prompt() function

Use the `proc.prompt()` function to display a prompt for a user to enter a value to the procedure.

Syntax

```
proc.prompt('question', 'hint');
```

| Parameter | Description |
|-----------|--|
| question | It specifies the question in the prompt to a user. |
| hint | It specifies the initial value of the parameter. |

Example

```
/*JavaScript*/
var tableName = proc.prompt('What table do you want to display?', 'Q.STAFF');
proc.exec('DISPLAY ' + tableName);
```

This example displays a prompt with the specified question and the hint that a user can modify and open the required table. In the example, the Q.STAFF table opens.

The proc.setCellValue() function

Use the setCellValue() function to edit a particular cell value in the current copy of the result set. The new edited result set can be used in SAVE or EXPORT command.

Syntax

```
setCellValue(row, column, value);
```

| Parameter | Description |
|-----------|--|
| row | It specifies the row of the cell, the value of which you want to get. |
| column | It specifies the column of the cell, the value of which you want to get. |

Example

```
/*JavaScript*/
proc.exec('DISPLAY Q.INTERVIEW');

var currentTime = new Date();
currentTime = currentTime.getHours() + ':' + currentTime.getMinutes()
+ ':' + currentTime.getSeconds();
var numRows = proc.getNumRows();
for (var i = 1; i <= numRows; i++)
{
    proc.setCellValue(i, 4, currentTime);
}
proc.exec('EXPORT DATA TO D:\qinterview_updated.txt')
```

The example procedure displays the Q.INTERVIEW query, gets the number of rows in the query result set, replaces time data in the fourth column with the current time, and exports the updated copy of the query results to the specified file: D:\qinterview_updated.txt.

Chapter 9. Managing query results

With query results in the available editor window you can perform several formatting tasks.

Settings for query results

You can customize the result set by modifying the row limit or resource limit.

About this task

You can modify the query settings for the result set by:

1. **Setting the Row Limit:** Set row limit for a query result for limiting the number of rows in the result set.
2. **Setting the Resource Limit:** Set the resource limit for a query result by configuring the row **Limits** for the user group, Large Objects (**LOB**) retrieval Options, enabling or disabling **Cache** for result sets, and setting **Acceleration** type for the analytics accelerator related queries.

Setting Row Limit

You can set row limit for a query result for limiting the number of rows in the result set.

About this task

Each time a user runs the query, the row limit controls the maximum number for rows fetched for the query.

You can set the row limit for the result set from the **Query** menu or by clicking the **Query Settings for Result Set** toolbar button. The **Query Settings for Result Set** button is enabled, when following conditions are satisfied:

- Connection to the data source is established
- Query has been run

To set the row limit, perform either of the following steps:

Procedure

1. From the **Query Settings for Result Set** toolbar button, select **Set Row Limit**. Enter the maximum number of rows to retrieve for this query and click **OK**.
Note: If the row limit is not specified or set to zero, no row limits are applied.
2. Alternately, you can select **Set Row Limit** from the **Query** menu and specify the maximum number of rows to retrieve.

Setting Resource Limit

You can set up resource limits that can be applied for a query.

About this task

You can set the resource limit for a query result by configuring the row **Limits** for the user group, Large Objects (**LOB**) retrieval Options, enabling or disabling **Cache** for result sets, and setting **Acceleration** type for the analytics accelerator related queries.

You can set the row limit for the result set from the **Query** menu or by clicking the **Query Settings for Result Set** toolbar button.

For the **Query Settings for Result Set** toolbar button to be enabled, following conditions must be satisfied:

- Connection to the data source is established
- Query has been run

Procedure

1. From the **Query Settings for Result Set** toolbar button, click **Set Resource Limit**.
2. In the **Limits** tab, populate the required information. For more information on **Limits**, see [“Specifying Limits resource limits”](#) on page 538.
3. In the **LOB Options** tab, populate the required information. For more information, see [“Specifying LOB Options resource limits”](#) on page 543
4. In the **Cache** tab, populate the required information. For more information, see [“Specifying Cache resource limits”](#) on page 544
5. In the **Acceleration** tab, populate the required information. For more information, see *Specifying Acceleration resource limits*.

This option is applicable only for analytics Accelerator related queries.

Formatting the query results

You can customize how the query results will appear in the editor window.

About this task

You can specify different fonts, colors and text alignments for each column header and each cell of the column. You can also specify that column and cell formatting be applied based on the results of a conditional expression.

Using the Properties view

From the **Properties** view you can quickly apply many formatting options to the query results.

About this task

To apply formatting options:

Procedure

1. At any time you can open the **Properties** view by selecting **Window > Show View > Properties**.
2. Select a cell or a column in the query results. The properties for the cell or column are listed in the **Properties** view.
3. To apply formatting, modify one of the properties that are listed in the in the **Properties** view.

Using the Layout Properties dialog box

On the **Layout Properties** dialog box you can set all the options that are available for formatting query results.

About this task

The **Layout Properties** dialog box uses a tree structure to represent the columns in the query results. Depending on what you have selected in the query results, the **Layout Properties** tree displays branches for all columns in the query results or for a single column in the query results. You can apply formatting options to entire columns, individual cells, column headers, and summary cells. The options are divided into the following groups:

- General

- Font
- Format
- Conditional

To apply formatting to the query results:

Procedure

1. Open the **Layout Properties** dialog box in one of the following ways:
 - Right-click in a column's header cell and select **Font** or **Format** from the pop-up menu.
 - Select **Results > Font**.
 - Select **Results > Format**.
2. The **Layout** tree opens differently depending on what you have selected from the query results. If you have not selected a specific column, then the tree displays all the columns in the query results. If you have selected a column, the tree displays the branch associated with that column. With only one or more columns selected and displayed, the **Show all columns** check box is available. Select this check box to display all the query results columns in the **Layout** tree. Clear the check box to display only the selected columns.
3. Click the plus sign to the left of the column name to expand the branch. For each column in the query results, the **Layout** tree has three branches. They are named **Header**, **Detail**, and **Grand Total**. The names of the branches are the same for each column. If expandable, there is plus sign to the left of each branch. Click the plus sign to expand the branch.
4. The level of the branch that you select in the **Layout** tree determines what formatting you can perform on the column of data. As you select each branch, the **Layout Properties** dialog changes. The formatting fields that can be used on the selected branch are made available.
5. To specify the formatting options that will apply to the entire column, select a column name branch. The **General** page opens within the **Layout Properties** dialog box. From this page you can specify a header name, column width, and summary row height.
6. To specify the font and formatting options that will be used for the column's header in the query results, select the **Header** branch for the column. The **Font** and **Format** pages open within the **Layout Properties** dialog box.
7. You can specify default and conditional formatting for the detail and summary cells of the column. Default formatting applies to all the cells in the column except for those that meet specific conditions. Conditional formatting applies to any cells in the column that meet specific conditions.
8. To specify the default font and formatting options that will be used for each cell in the column, expand the **Detail** branch for the column, then select the **Default** branch. The **Font** and **Format** pages open within the **Layout Properties** dialog box. Select the **Font** tab to specify the formatting options that will apply to the font that is used to display the data in the column's detail cell. Select the **Format** tab to specify how the data will be formatted in the detail cell of the column.
9. To specify conditional formatting options for a column's detail cells, expand the **Detail** branch for the column, then select the **Default** branch or an existing condition branch. The fields used for specifying conditional formatting become available. Specify the condition name and expression. You can add one or more conditions for the detail cells.
10. To specify the default font and formatting options that will be used for a column's summary line, expand the **Grand Total** branch, then select the **Default** branch for the column. The **Font** and **Format** pages open within the **Layout Properties** dialog box. Select the **Font** tab to specify the formatting options that will apply to the font that is used to display the data in the column's summary line cell. Select the **Format** tab to specify how the data will be formatted in the summary line cell of the column.
11. To specify conditional formatting options for a column's summary cell, expand the **Grand Total** branch for the column, then select the **Default** branch or an existing condition branch. The fields used for specifying conditional formatting become available. Specify the condition name and expression. You can add one or more conditions for the summary line cell.

12. Click **Apply**. The **General**, **Font**, or **Format** options that you have specified are saved.
13. You can repeat this process to set up unique formatting options for each column in the query results. You can also duplicate the options that you have specified for one column and apply them to other columns using either of the following buttons:
 - **Apply to all columns**: Use this button to apply the **Font** or **Format** options that you have specified to all the columns that are currently selected from the query results. For example, if you specify special font options for the **Detail Default** branch of a column, and select the **Apply to all columns** button, the font options that you have specified will be applied to the **Detail Default** branch of each column that is currently selected and displayed in the **Layout Properties** tree.
 - **Apply to all levels**: Use this button to apply the **Font** or **Format** options that you have specified for the current level of the column to all levels of the column. For example, if you specify special font options for the **Detail Default** branch of a column and select the **Apply to all levels** button, the font options that you have specified will be applied to the **Grand Total Default** branch for the column.
14. Click **OK**. The **Layout Properties** dialog box closes. The format information that you have specified is saved and applied to the columns in the query results.

Specifying general column formatting options

On the **General** page of the **Layout Properties** dialog you change the header text for a column, the column width, or the height of a row. In addition, you can see the grouping or aggregation value that has been applied to a column and the name of the column.

About this task

General formatting options apply to all cells in a query results column including the header cell, each detail cell, and the column's summary line cell.

To specify the **General** formatting options:

Procedure

1. Open the **Layout Properties** window and click a **column name** branch. The **General** page opens within the **Layout Properties** window.
2. The **Heading** field displays the name of the column as it appears in the query results. You can change the header text by typing a new name in this field.
3. The **Grouping and aggregation** field displays the aggregation codes that have been specified for the column. If none have been specified, this field is not blank. This field is informational and cannot be modified.
4. The **Width** field specifies the width of the column in pixels. You can change the width by typing a new number in the field.
5. The **Summary row height** field specifies the height of the summary column (if aggregation is selected) in pixels. You can change the height by typing a new number in the field.
6. The **Source** field identifies the column that is currently selected from the **Layout Properties** tree. If the selected column is a calculated column, the conditional expression that was used to create the results in the column is displayed. This is an informational field and cannot be modified.
7. Click **Apply** to save the general formatting options that you have specified.

Specifying font formatting options

On the **Font** page of the **Layout Properties** window, you can specify unique font and color attributes for each column's header cell, for each detail cell in every column, for each break point summary cell (if break columns have been specified), and for the column's final summary cell.

About this task

To specify the font formatting options:

Procedure

1. Open the **Layout Properties** window. Click the **Font** tab.
2. To specify the font attributes for the header cell of the column, select the **Header** branch under the column name.

Specify the font attributes that you want for the column's header cell in the **Font**, **Font style**, **Size**, **Foreground**, and **Background** fields.

3. To specify the font attributes for the detail cells of the column, select the **Detail** branch under the column name.
 - a) From the **Detail** branch, select the **Default** branch to specify the font attributes that will be used for all detail cells in the column.
 - b) Select a **Condition** branch to specify the font attributes that will be used for a detail cell that meets a specific condition.
 - c) You can create multiple conditional statements for the detail cells in the column.

Once you have selected the **Default** or a **Condition** branch, specify the font attributes that you want applied to the qualifying detailed cells in the **Font**, **Font Style**, **Size**, **Foreground color**, and **Background color** fields.

4. To specify the font attributes for the summary cell of the column, select the **Grand Total** branch under the column name.

Summary cells are not always visible in the query results column. The summary cell is visible only if you have specified a grouping and aggregation code for the column.

- a) From the **Grand Total** branch, select the **Default** branch, to specify the font attributes that will be used for the summary cell.
- b) Select a **Condition** branch to specify the font attributes that will be used for the summary cell if the value of the cell meets a specific condition.

You can create multiple conditional statements for the summary cell in the column. Once you have selected the **Default** or a **Condition** branch, specify the font attributes that you want in the **Font**, **Font Style**, **Size**, **Foreground color**, and **Background color** fields.

5. Click **Apply** to save the font attributes that you have specified.

The formatting information that you have specified for the column is applied in the query results.

What to do next

You can continue formatting the query results columns or click **OK** to close the **Layout Properties** window.

Font fields

Descriptions for each of the font attributes that can be applied to a query result column's header, detail and summary cells.

Font

The name of the font, such as Arial or Times Roman.

Font style

The style of the font, such as bold or italic.

Size

The point size of the font.

Script

The type of script being used, for example Western.

Foreground color

The color of the text, such as dark blue.

Background color

The color the text is against, such as a gray background.

Sample

A rendering of the current formatting.

Set as default

Sets the current font attributes as the default for subsequent query results. The default will remain in effect until you choose a new font or change the default.

Reset to default

Resets the font formatting attributes to the default setting.

Specifying formatting options

On the **Format** page of the **Layout Properties** window, you can specify unique attributes that will control how the data is positioned and displayed (i.e. number of decimal positions) in the query results columns.

About this task

You can specify different format attributes for each column's header cell, for each detail cell in the column, for each break point summary cell (if break columns have been specified), and for the column's final summary cell. The available **Format** options vary depending on whether you selected a column with character, numeric, date, or time data.

To specify data format options:

Procedure

1. Open the **Layout Properties** window. Click the **Format** tab.
2. To specify format attributes for the header cell of the column, select the **Header** branch under the column name. Specify the format information that you want for the column's header cell in the **Format**, **Horizontal Alignment**, **Vertical Alignment**, and **Options** fields.
3. To specify format attributes for the detail cells of the column, select the **Detail** branch under the column name. From the **Detail** branch, select the **Default** branch, to specify the formatting that will be used for all detail cells in the column. Select a **Condition** branch to specify the formatting that will be used for a detail cell that meets a specific condition. You can create multiple conditional statements for the detail cells in the column. Once you have selected the **Default** or a **Condition** branch, specify the formatting that you want in the **Format**, **Horizontal Alignment**, **Vertical Alignment**, and **Options** fields.
4. To specify the formatting for the summary cell of the column, select the **Grand Total** branch under the column name. Summary cells are not always visible in the query results column. The summary cell is visible only if you have specified a grouping and aggregation code for the column. From the **Grand Total** branch, select the **Default** branch to specify the formatting that will be used for the summary cell. Select a **Condition** branch to specify the formatting that will be used for the summary cell if the value of the cell meets a specific condition. You can create multiple conditional statements for the summary cell in the column. Once you have selected the **Default** or a **Condition** branch, specify the formatting that you want in the **Format**, **Horizontal Alignment**, **Vertical Alignment**, and **Options** fields.
5. Click **Apply** to save the format attributes that you have specified. The formatting information that you have specified for the column is applied in the query results. You can continue formatting the query results columns or click **OK** to close the **Layout Properties** window.

Format fields

Descriptions for each of the format attributes that can be applied to a query result column's header, detail and summary cells.

Format

- **As is** can be applied to numeric, character, date and time data.
- **Currency** can be applied only to numeric data.

- **Decimal** can be applied only to numeric data.
- **Scientific** can be applied only to numeric data.
- **Percent** can be applied only to numeric data.
- **User Currency** can be applied only to numeric data.
- **Hexadecimal** can be applied only to character data.
- **Binary** can be applied only to character data.
- **YYYYxMMxDD** can be applied only to date data.
- **MMxDDxYYYY** can be applied only to date data.
- **DDxMMxYYYY** can be applied only to date data.
- **YYxMMxDD** can be applied only to date data.
- **MMxDDxYY** can be applied only to date data.
- **DDxMMxYY** can be applied only to date data.
- **HHxMMxSS** can be applied to time data.
- **Custom** can be applied to different data types.

Sample

Shows a snapshot of how the data will be formatted.

Horizontal alignment

From the drop-down list, select how to align data horizontally in the column. Options are:

- **Left**
- **Right**
- **Center**

Vertical alignment

From the drop-down list, select how to align data vertically in the column. Options are:

- **Top**
- **Center**
- **Bottom**

Options

- **Thousands separator** can be applied to numeric data. Select this check box to place a thousands separator in numeric values (the symbol is taken from operating system settings).
- **Negative numbers** can be applied to numeric data. Select the format for a negative number from the list of available options.
- **Decimal places** can be applied to numeric data. Specify the number of decimal places that will be included in the numeric value.
- **Wrap text** can be applied to character data. Select the check box to wrap the contents of the cell to the next line within the cell.
- **Separator symbol** can be applied to date and time data. From the list of available formats, select one that will be used to format the date or time data. For example, select a slash (/) for date data to format the data as: 03/25/02. Select a symbol to format time data, such as a colon (:). An example is: 12:03:02.
- **12 hour mode** can be applied to time data. Select the check box to display time in a 12-hour interval, such as 03:12:30 PM or 10:05:07 AM. If this check box is cleared, time is displayed in 24-hour format (such as 15:12:06).

Set as default

Sets the current format attributes as the default for subsequent query results. The current settings will remain in effect until you change the default.

Reset to default

Resets the format attributes to the default setting.

The following formats are available for date:

- **YYYYxMMxDD**
- **MMxDDxYYYY**
- **DDxMMxYYYY**
- **YYxMMxDD**
- **MMxDDxYY**
- **DDxMMxYY**

The following formats are available for time:

- **HHxMMxSS**
- **HHxMM**

The following formats are available for timestamp:

- **YYYYxMMxDD HHxMMxSS**
- **DDxMMxYYYY HHxMMxSS**
- **YYYYxMMxDD HHxMM**
- **DDxMMxYYYY HHxMM**
- **YYxMMxDD HHxMMxSS**
- **DDxMMxYY HHxMMxSS**
- **YYxMMxDD HHxMM**
- **DDxMMxYY HHxMM**

Specifying conditional formatting options

Conditional formatting options can be set for an entire column of data in the query results or for each detail, summary and grand total cell in a column. Based on the evaluation of a conditional expression, each column and cell in the query results can have a unique set of formatting attributes.

About this task

For example, within a column, you could specify that if a cell's content equals 0, then display the cell with a red background and if the value is 100, display the cell with a blue background.

To set conditional formatting:

Procedure

1. Open the **Layout Properties** window. You can specify conditional formatting from either the **Font** page or the **Format** page depending on what results you want to apply to the data in the cell or column.
2. Apply conditional formatting.
 - a) Click the **Detail** branch to apply conditional formatting to the detail cells of the column.
 - b) Click the **column name Total** branch to apply conditional formatting to the summary cell of the column.
 - c) Click the **Grand Total** branch to apply conditional formatting to the grand total cell of the column.
3. With the appropriate branch selected, click the **Add** icon. The control fields **Condition name** and **Condition Expression** become available.
4. Provide a unique name to each conditional expression that you create by typing the name of the expression in the **Condition name** field.

A branch is automatically created in the tree using the condition name.

Each condition must have an expression.

5. Type the expression in the **Condition Expression** field.

You must follow the rules for building conditional formatting expressions. The expression language supports constant expressions, expressions defined as variables sourced on other columns in the result set (1, 2, for example), numeric operators (such as +, -, *, /), character operators (+, CONCAT), numeric and character functions, and a set of logical and relational operators (such as &&, ||, !=, ==).

6. After you have created the expression, select the **Font** or the **Format** page of the **Layout Properties** window to specify the formatting that will be applied to any cell in the column that meets the conditions set in the expression.


Filtering query results

You can apply sorting and filtering options to the columns or your query result set to control the rows that they include.

About this task

To apply sorting and filtering options to one or several columns or your query result set, complete the following steps for each of the columns that you want to filter:

Procedure

1. On the **Results** tab, click  on a column whose values you want to sort or filter.

Note: If your query result set contains top groups or side groups, you cannot apply filters to it.

2. To sort the column values in an ascending order, click **Sort Ascending**.
3. To sort the column values in a descending order, click **Sort Descending**.
4. To remove the currently applied column filter, click **Remove Filter**.

Note: The **Remove Filter** function does not allow you to remove free style condition filters and prompt hierarchy filters.

5. To open the **Dynamart Filter** wizard that allows you to configure advanced filtering options, click **Advanced**. For information about using the **Dynamart Filter** wizard, see the [“Filtering query results by using the Dynamart Filter wizard”](#) on page 187 topic.

Note: Free style condition filters and prompt hierarchy filters are applied to a query result set instead of individual columns. If you switch from a free style condition filter or a prompt hierarchy filter to a compound or simple filter, the free style condition/prompt hierarchy filter is removed.

6. In the **Type filter text** field, type the text by which you want to filter the column list.
7. From the column list, select the row values that you want to include into your query result set.
8. If the column list does not include all the row values from your query result set, click **Not all values are fetched** to display the remaining rows values.

Note: The maximum number of rows that can be fetched is 10000.

9. Click **OK** to apply your filter.

Filtering query results by using the Dynamart Filter wizard

You can use the **Dynamart Filter Wizard** to create filters that control the contents of the query result set.

About this task

Filters allow you to restrict the returned rows and values of a query result set to your specifications. Any filter can contain any number of expressions that give you complete control over the content of the grid. To filter query results.

Procedure

1. Open the query result set that you want to filter in the **Results** editor.
2. Select **Results > Filter Results**.
The **Dynamart Filter Wizard** opens.
3. Select the **Enable filtering** check box.
From here, you can choose to build your filter one of three ways: use a table of options to assist you in building your filter, use the expression designer to build a free style filter, or use an embedded prompt hierarchy.
4. Select a filter building method and click **Next**.
The next page of the wizard depends on your previous selection.
5. Build your filter using the table of options or the expression designer and click **Finish**.
The **Dynamart Filter Wizard** closes and the filters that you specified are applied to the query results.

Building a compound condition filter

A compound condition filter allows you to filter query results using multiple conditions.

About this task

Use a compound condition filter to set multiple conditions to filter your query results. You can add more than one condition from a table of options in the **Dynamart Filter Wizard**. To build a compound condition filter:

Procedure

1. Open the **Dynamart Filter Wizard**, select the **Enable filtering** check box, select **Compound condition filter** and click **Next**.
The **Edit dynamart filter** page opens.
2. Select the column that you want to filter from the **Column** drop-down list.
3. Select the operator of the filter from the **Operator** drop-down list.
Valid options include:
 - **Not Less Than** - All values that are equal to or greater than the specified value are included.
 - **Greater Than** - All values that are greater than the specified value are included.
 - **Not Greater Than** - All values that are equal to or less than the specified value are included.
 - **Less Than** - All values that are less than the specified value are included.
 - **Equal To** - Only values that are equal to the specified value are included.
 - **Not Equal To** - All values that are not equal to the specified value are included.
 - **Like** - All values that match the specified pattern are included. The pattern can contain the following special characters:
 - % matches zero (0) or more characters.
 - _ matches only one (1) character.
 - ? matches only one (1) character.
 - **Case-Insensitive Like** - A case-insensitive version of the Like operator.
 - **Unlike** - All values that are not like the specified value are included.
 - **Is Null** - Only null values are included.
 - **Is Not Null** - All of the values that are not null are included.
 - **In Range** - All of the values that fall within the two specified values, along with the values themselves, are included.
 - **Outside Range** - All of the values that fall outside of the two values, along with the values themselves, are included.

- **Between** - All of the values that fall within the two specified values are included.
 - **Not Between** - All of the values that fall outside of the two specified values are included.
 - **In** - One or several values that are specified by a user are included. To filter your query result set by several distinct values, you must create a simple prompt of **Enumeration** type, select **Prompt** from the **Type1** list, and then select your prompt from the **Value1** list. If you select **Column** from the **Type1** list, the result set will include the rows that contain matching values in both columns. If you select **Constant** from the **Type1** list, the result set will include the rows that contain the specified value.
4. Select the type of value that you want to specify from the **Type1** list.
Valid options include:
 - **Constant** - The specified value is a set constant, such as a numeral or character.
 - **Column** - The specified value is another column in the result set.
 - **Prompt** - The specified value or values are provided by a simple prompt or a prompt hierarchy that is associated with the query.
 5. Specify the value of the expression in the **Value1** column.
 6. If you selected an applicable operator type, specify a type and value in the **Type2** and **Value2** columns.
 7. If you have multiple expressions listed, select the condition from which the expressions are run from the **Condition** drop-down list.
 8. Click **Move Up** or **Move Down** to change the order in which the different expressions are run.
 9. Click **Down Level** or **Up Level** to change the priority level of each expression.
An expression with a lower level number has more priority than an expression with a higher level number.
 10. To remove an expression, select it and click **Remove**.
 11. Once you are done building and arranging your expressions, click **Finish**.
The **Dynamart Filter Wizard** closes and the query results are filtered.

Building a free style condition filter

A free style condition filter allows you to filter query results using the expression designer.

About this task

One way to filter your query results is to build a set of expressions from the expression designer. This method is recommended for anyone who is familiar with the QMF expression designer syntax. To build a free style condition filter:

Procedure

1. Open the **Dynamart Filter Wizard**, select the **Enable filtering** check box, select **Free style condition filter** and click **Next**.
The **Edit dynamart filter expression** page opens.
2. Using proper expression syntax, write the expression that you to use to filter the results in the **Expression** field.
3. Use the function categories on the right to help you construct your filter expressions.
Note: You can use you own functions to construct the filter expression. For more information, see *Adding custom functions*.
4. Once you have finished building you filter expressions, click **Finish**.
The **Dynamart Filter Wizard** closes and the query results are filter.

Building a prompt hierarchy filter

A prompt hierarchy filter allows you to filter query results using a prompt hierarchy that is used in the query.

About this task

Use the Dynamart Filter wizard to build a prompt hierarchy filter.

Procedure

1. Open the **Dynamart Filter Wizard**, select the **Enable filtering** check box, select **Prompt Hierarchy filter** and click **Next**.
2. On the **Assign prompt hierarchy levels to columns** page, use the **Prompt** column to select the prompt hierarchy level for the column.
3. Once you have finished building the filter, click **Finish**.
The **Dynamart Filter Wizard** closes and the query results are filtered.

Working with calculated columns

Calculated columns are columns of data that you can add to the query result set.

Having access to the appropriate syntax reference information and function reference information before using the calculated column feature will help you to build expressions more quickly.

For information about syntax, see *Syntax for calculated column expressions* in the reference section of the User help.

For a list of all available functions, including the functions for calculated columns, see *Functions for calculated columns* in the reference section of the User help.

Adding calculated columns to a query result set

You use the Calculated Columns window to add a calculated column to the query result set. You can add a calculated column before or after existing columns.

About this task

The content of an added column is calculated by using an expression that you define. The expression can use data from other columns and constants. The expression used to calculate the value of the added column is saved with the query. The calculated column will be included in the query result set each time the query is run.

To add a calculated column to the query result set, perform the following steps:

Procedure

1. Open the **Calculated Column** window in one of the following ways:
 - If you are using QMF for Workstation, on the **Results** page of the editor, right-click a column's header in the query results and select either **Add Calculated Column Before** or **Add Calculated Column After** from the pop-up menu.
 - With query results in the active editor, select **Results > Add Calculated Column**.
2. In the **Calculated Column** window, specify the name of the calculated column as it should appear in the query results in the **Name** field.
3. Select the type of the calculated column from the **Type** list. The type determines the format of the calculated column data.
Valid choices are:
 - **Integer**: Data is expressed as whole numbers. For example, "5" or "100".

- **Decimal:** Data is expressed as a real number. For example, "10.5" or "0.25".
 - **Character:** Data is expressed only as specific text strings. For example, "John Smith", or "10 Main Street".
 - **Date:** Data is expressed in a date format. For example, "12/03/2000" or "1/2/09".
 - **Boolean:** Data is expressed as a Boolean value. For example, "true or false", or "yes or no".
 - **Categorization:** Data is represented by category names that are defined for a range of data values. For example, "small, average and large" can be used to categorize sales quantities that range between 1,000 and 100,000 units.
4. In the area containing all available functions, navigate to and select the required one. You can use the filter to quickly find the function you need.
 5. If you want to load functions from JavaScript Modules, click the **Add JavaScript Modules** button and add the modules with functions in the **JavaScript Modules** window. The loaded functions become available in the **JavaScript Modules** node.

Tip: You can delete JavaScript modules from queries in the **JavaScript Modules** window.
 6. Build the expression

In the **Expression:** window, enter the syntax for the expression to be used to calculate the contents of the column.

The expression that you build can use data from other columns, constants and functions. You must adhere to a specific set of syntax rules when building the expression for the calculated column.

The expression language supports the following rules:

- constant expressions defined as variables sourced on other columns (column 1, column 2, for example) in the result set
- numeric operators (+, -, *, /), character operators (+, CONCAT)
- numeric and character functions (MIN, MAX, AVG, SQRT, CONCAT, SUBSTR)
- a set of logical and relational operators (<, =, !=, and others).

The expression used to calculate the value of the added column is saved with the query.

7. Click **OK** to close the **Calculated Column** window.

Results

The content of the column is calculated according to the specified expression and added to the query result set.

If you selected **Add calculated column** from the pop-up menu, the new column is added to the as the last column in the query results.

If you selected **Add calculated column before** from the pop-up menu, the new column is inserted in the query results before the selected column.

If you selected **Add calculated column after** from the pop-up menu, the new column is inserted in the query results after the selected column

Adding custom functions

You can add JavaScript functions to QMF for Workstation and QMF for WebSphere and use them for building different expressions, for example, when adding a calculated column or creating a dynamart filter.

About this task

You can create the `functions.js` file or use JavaScript modules from repository to add custom JavaScript functions.

The following topics describes the detailed procedures of adding custom functions:

Adding custom functions from a file

You can create a file with JavaScript functions and add them to QMF for Workstation and QMF for WebSphere. Use the functions for building different expressions, for example, when you add a calculated column or create a dynamart filter.

About this task

To create and add a custom function:

Procedure

1. Create a file called *functions.js* with the description of JavaScript functions and their implementation.
2. Start each function with the keyword 'function'.
3. Add a comment before each function.

The application interprets this comment as the description of the function.

4. Optional: Define a category to which you add a new function.

Use the '@category' tag in the comments to add it. If there is no such category, it will be created.

Note: The following categories already exist in the application:

- Columns
 - Aggregation
 - Conversion
 - Date and Time
 - Information
 - Logical
 - Math and Trigonometric
 - Misc
 - Text
 - Objects
5. Place the *functions.js* file in the QMF for Workstation or QMF for WebSphere user home directory.
If you want to share *functions.js* with other users, they will have to copy the *functions.js* file to their home directories.
 6. Launch or restart your application.

Results

The function is added to the defined category and can be used for building expressions for calculated columns, dynamart filters, and scheduler.

This allows you to implement various features in other QMF objects such as visual dashboards. For example, you can display data retrieved from a table on a map according to address information from that table. For more information, see [“Displaying data on a map according to address information retrieved from a query”](#) on page 394.

Adding custom functions from repository

You can use the functions from JavaScript modules for building different expressions in calculated columns.

Before you begin

You must have a JavaScript module with functions in the repository. For more information about JavaScript modules, see [“Creating JavaScript Modules”](#) on page 398.

About this task

To add functions from JavaScript modules:

Procedure

1. Open the query to that you want to add a JavaScript module with the functions in the query editor.
2. From the main menu, select **Query > Manage JavaScript Modules**. The **JavaScript Modules** window opens.
3. Click the **Add JavaScript Module** button, the **Open from Repository** window opens.
4. Select the JavaScript module that you want to add, click **Finish**.

The module is added and the available functions are listed in the **Available Functions** list.

Results

You added a JavaScript module to a query and can use the module functions in calculated columns. The functions from the added JavaScript modules are available in the **JavaScript Modules** node of the **Calculated Column** window.

Applying categorization to query results

You can apply categorization to the calculated columns in results of visual queries. Use categorization to stratify the data and organize it into categories for effective work.

Before you begin

Open a query to which results you want to apply categorization on the **Design** tab of the Visual Query editor.

About this task

To add categorization:

Procedure

1. In the **Transformations** pane, click the **New Categorization** button to open the **Categorization** dialog.
2. Enter the name that you want to use for the categorization in the **Categorization Name** field.
3. Click **Add Conformance Rules Set** to open the **Conformance Rules** dialog and specify the rules that you want to use to include the appropriate values of the result set in the category.
4. Click **Add Category** to open the **Category** dialog and specify the expressions for categorization.
5. In the **Categorization** dialog, select the presentation for the values that are not included in the category.
 - Select **Left Out** to display empty values.
 - Select **Grouped Together** and in the **Group name** field specify the name that you want to display for all values that are not included in the category.
 - Select **Listed Individually** to display the values of the result set.
6. Click **OK** to close the **Categorization** dialog.

The categorization that you created is added to the **Categorizations** node in the **Transformations** pane.

Results

You can use the categorizations that you created to display specific data in the calculated columns of the query results.

Adding conformance rules for categorization

You specify conformance rules to include the appropriate values of the result set in the category.

About this task

To add conformance rules:

Procedure

1. In the **Categorization** dialog, click **Add Conformance Rules Set** to open the **Conformance Rules** dialog.
2. In the **Category Name** field, enter the name that you want to use for the result set values that are included in the category.
3. Click **Add** to add a rule to the **Rules of Category** field.

Note: You can add multiple rules to the same category.

4. Select the type of the rule from the **Type** area.
5. To specify the condition under which the rule is executed, select the operator from the **Operator** list and specify the required values in the **Value1** and **Value2** fields.

Note: The availability of the **Value2** field depends on the operator that you selected.

6. To change the order of the specified rules, use the **Up** and **Down** buttons.
7. When you finished specifying the rules, click **OK** to return to the **Categorization** dialog.

Related tasks

[Adding expressions for categorization](#)

You specify expressions to categorize the values of the result set.

Adding expressions for categorization

You specify expressions to categorize the values of the result set.

About this task

To add expressions for categorization:

Procedure

1. In the **Categorization** dialog, click **Add Category** to open the **Category** dialog.
2. In the **Category Name** field, enter the name that you want to use for the result set values that are included in the category.
3. In the **Expression for rules** field, specify the expression that you want to use to categorize the values of the result set. Use the expression designer functions that are on the right of the **Expression for rules** field.
4. When you specified the expression, click **OK** to return to the **Categorization** dialog.

Related tasks

[Adding conformance rules for categorization](#)

You specify conformance rules to include the appropriate values of the result set in the category.

Applying grouping and aggregation to query result columns

Grouping and aggregation options can be applied to query result columns to organize the result data into logical or summarized groupings.

About this task

By adding grouping and aggregation you can automatically obtain summary information about your data and display the data more logically. For example you can roll up data in a report by department or average departmental commissions.

To apply grouping and aggregation to the query result columns:

Procedure

1. Access the menu of grouping and aggregation options in one of the following ways:
 - With query results in the available editor, select **Results > Grouping and Aggregation**.
 - If you are using QMF for Workstation, on the **Results** page of the editor, right-click a column's header cell in the query results and select **Grouping and Aggregation** from the pop-up menu.
 - From the **Display** page of the editor, right-click a column name in the **Layout structure** tree and select **Grouping and Aggregation** from the pop-up menu.
2. Select the grouping and aggregation option that you want to apply to the column.
3. Repeat the process for each column of data in the query results to which you want to add grouping and aggregation.

Grouping and aggregation fields

Grouping and aggregation options can be applied to query result columns to organize the result data into logical or summarized groupings.

No aggregation

Select this option to remove any aggregation or summary formatting from the selected column.

Top Group

Select this option to group data across the top of the query results by the selected column. Placing a column in a top group will display related values of that column on the horizontal axis, which is known as *pivot* or *ACROSS* functionality. You can define multiple columns of top groups. You can specify that **Top Groups** include summary information. With the column selected, click **Show Total Summaries** from the pop-up menu. The query results will be reformatted so that the column spans the other columns in the result set. A summary line for each row in the query results is added.

Side Group

Select this option to group data down the left side of the query results by the selected column.

Count

Select this option to create a summary row at the bottom of the query results showing the number of values in the column.

First

Select this option to create a summary row at the bottom of the query results showing the first value in the column.

Last

Select this option to create a summary row at the bottom of the query results showing the last value in the column.

Maximum

Select this option to create a summary row at the bottom of the query results showing the maximum value in the column.

Minimum

Select this option to create a summary row at the bottom of the query results showing the minimum value in the column.

Sum

Select this option to create a summary row in the query results showing the sum of the values in the column. This option calculates interim summaries and overall summaries if another column is selected for grouping with summaries.

Cumulative sum

Select this option to display the values in the column as a cumulative sum, which is a rolling sum. Each value is added to the value in the row below it and the sum of the two values becomes the value that is added. A cumulative summary is presented at the end.

Average

Select this option to create a summary row showing the average of the values in the column.

Standard Deviation

Select this option to create a summary row showing the standard deviation of the values in the column. The standard deviation is a statistic that tells you how tightly all the various examples are clustered around the mean in a set of data.

Percentage of group

Select this option to display the values in the column as a percentage of the group's total, with a summary row showing the sum of the percentages. **Percentage of group** identifies how each value in the group contributes to the whole. For example a salary of \$1,000 is 10% of a total salary of \$10,000.

Percentage of total

Select this option to display the values in the column as a percentage of the column total, with a summary row showing the sum of the percentages.

Cumulative percentage of group

Select this option to display the values in the column as a percentage of the group total. A summary row at the bottom of the query results shows the sum of the percentages. For example, DEPT 42 earns 0.42% in commissions.

Cumulative percentage of total

Select this option to display the values in the column as a percentage value of the group total and a percentage value of the column total. A summary row at the bottom of the query results shows the sum of the percentages. For example, within DEPT 42, Wheeler earns 44.60% of the commissions, while Williams earns 55.40%, and Smith earns no commission. Overall, this DEPT 42 earns 5.75% in commissions for the company.

Exporting query results

You can export query results data to a database or to a file.

About this task

You use the **Export Query Results** wizard to export query results.

To export query results:

Procedure

1. Open the **Export Query Results** wizard in one of the following ways:
 - With a set of query results in available editor, select **Results > Export**.
 - Select **File > Export**. Select **Result set** from the list of available export wizards.
2. The **Export Query Results** wizard opens with the **Export query results** page.

If you opened the **Export Query Results** wizard using **Results > Export**, the name of the active query results that you will export is listed in the **Source** field.

If you opened the **Export Query Results** wizard using **File > Export**, then you must select the set of query results that you will export from the **Source** field.

Note: If the query that you are exporting includes several result sets, all of them are displayed in the **Source** field.

3. Select where you want to export the query result data by choosing one of the following:
 - Click **File system** to export the query results data to a file. The fields that are required to export query results to a file system are displayed.
 - Click **Database** to export the query results data to a database. The fields that are required to export query results to a database are displayed.

Sending query results in an email

You can email query results using the **Internet Mail Wizard**.

About this task

This task describes how to send query results in an email.

Procedure

1. With a set of query results in active editor, select **File > Send To > Mail Recipient...**

Note: If your query contains several result sets, only the currently active display mode is sent.

The Export Query Results wizard opens with the **Select an export file type** page.

2. Select from the file types that display and click **Next**.

- Text - Delimited Text (*.txt)
- HTML – Hypertext Markup Language (*.htm)
- IXF - Integrated Exchange Format (*.ixf)
- CSV – Comma-separated values (*.csv)
- DBF - dbase III (*.dbf)
- XML - QMF Markup Language file (*.xml)
- WQML - Web Query Markup Language (*.wqml)
- PDF - Portable Document Format (*.pdf)
- XLS - Microsoft Excel 97-2003 (*.xls)
- XLSX - Microsoft Excel 2007 (*.xlsx)

The second page of the **Export Query Results** wizard opens.

3. Select additional options for the file type and click **Finish**.

Note: Available options depend on the file type you have selected. For more information, see *Specifying file export options*.

The **Internet Mail Wizard** opens.

4. On **Message parameters** page, specify values for the following parameters:

- Select the format of the body text from the **Format** list. The available formats are Text and HTML.
- The address to that the email must be sent in the **To** field.
- The address to that a carbon copy of the email must be sent in the **Cc** field.
- The subject of the email in the **Subject** field.
- The body text of the email in the **Message** field.
- The address from that the email must be sent in the **From** field.

5. To specify SMTP server parameters, click **SMTP Settings**. In the **SMTP Settings** window, specify whether to use the server that is predefined by administrator or a custom one.

- a) If you decide to use a predefined SMTP server, select the server from the list of predefined servers or click **Add Server** to create one.
- b) If you decide to use a custom server, specify server parameters: its address and port number. If the SMTP server requires authentication, select **Use SMTP authentication**, and then specify user name and password. If it is necessary to specify extra access preferences, click **Advanced** to open the **Advanced SMTP Settings** window where you can add keywords that are supported by `javamail`. Click **OK**.

Click **OK** to close the **SMTP Settings** window.

6. Click **Next** to add another attachment
7. Click **Click here to add new file** label
8. Click the ellipsis (...) and select the necessary file from your hard disk drive.

You can attach images as well as files of other available types.

If you want to embed an image in the body text of the email, you must add the following HTML-tag to the area of the text in which you want the image to display:

```
</img>
```

Where `cid:[image name]` identifies the image file you are embedding. For example, if you wanted to embed a `.gif` file named `chartx`, you would insert the following HTML-tag into the text:

```
</img>
```

9. Click **Finish** to send the message.

Exporting query results to a database

You can export query results data to a database using the **Export Query Results** wizard.

About this task

To export query results data to a database:

Procedure

1. To open the **Export Query Results** wizard, select **File > Export**. Select **Result set** from the list of available export wizards and click **Next**.
2. Select **Database**. The fields that are required to export query results to a database are displayed.
3. From the **Source** list, select the query result set that you want to export.
Note: If the query that you are exporting includes several result sets, all of them are displayed in the **Source** field.
4. Specify the database where you want to save the query results data in the **Database** field. When you select a database, the **Table space** field is updated to reflect the table space names that can be used.
5. Specify the owner of the table where you want to save the data in the **Table owner** field. Specify the name of the table where you want to save the data in the **Table name** field. You must use the correct format for identifiers when entering the owner or name of the table.
6. Optional: If you want to attach a comment to the data, type the comment in the **Comment** field.
7. Specify a table space name.

A default table space name that has been specified in the resource limits for your user ID is displayed when you select a database. This table space name will be used unless you specify another table space name in the **Table space** field.

You can only specify a new table space name if the table that you are adding data to does not already exist. In addition, the resource limits for your user ID must grant you permission to specify or change the table space name.

8. Click **Next**. The **Set up save options** page opens.

9. You can specify how much of the data in the current query results will be exported by selecting one of the following:
- **All** to specify that all of the data in the current results grid will be saved with all the changes applied to it. For example, if you remove a column from the layout, it will not be saved in the export file.
 - **All - Result set only** to specify that all of the data in the original query result set not depending on the changes applied to results grid will be saved. For example, if you remove a column from the layout, it will be saved in the export file anyway.
 - **Selection** to specify that only the data that you have selected in the current query results will be saved. This can include calculated columns if they are selected. The query result data that you want to save must be selected before you open the **Export Query Results** window.
- Note:** The **Selection** option is disabled when the fast save mode is selected in the next step.
- Note: All - Result set only** is the only available option when you export a query result set from a forecast.
10. Specify the method by which the data is saved:
- Select **Regular - Send retrieved data back to data source and save (using an insert statement for each row)** to select the regular save mode.
 - Select **Fast - Directly save results at the data source by re-running the query at the data source** to select the fast save mode that retains any ORDER BY clause(s).
- You can select the **Remove ORDER BY clause(s) from original SQL before re-running query** check box to remove the ORDER BY clause(s) from the query.
- Note:** Not all database servers are able to perform a fast save of data. Check with your system administrator if you receive errors when saving with this method.
11. To export any calculated columns that have been added to the query results, make sure the **Save calculated columns** check box is selected.
- Note:** This option is not available when:
- **All - Result set only** is selected.
 - **Selection** is selected without any of the calculated columns selected.
 - **Fast - Directly save results at the data source by re-running the query at the data source** is selected.
12. Specify how existing data is to be handled by selecting one of the following:
- **Replace any existing data** to replace any existing data that is in the specified table on the database.
 - **Add to any existing data** to add to any existing data that is in the specified table on the database.
13. Specify a value in the **Commit scope** field that will represent the number of rows to insert before committing changes. If this value is omitted or zero, all of the rows are inserted before a commit occurs.
14. After setting up the save options, click **Next** to go the page where you can set up the columns of the exported result set.
- Note:** Click **Finish** to skip the next step and close the **Export Query Results** wizard.
15. You can set up the columns by using following options:
- In the **Type** column, select a data type from the list.
 - In the **Length** field, specify a maximum number of characters in a string.
 - In the **Precision** field, specify a maximum total number of digits for the DECIMAL type.
 - In the **Scale** field, specify a maximum number of fractional digits for the DECIMAL type.
16. Click **Finish** to export the query result data. The **Export Query Results** wizard closes.

Exporting query results to a file

You can export query results data to a file system using the **Export Query Results** wizard.

Before you begin

To export a query result set to Microsoft Excel correctly, start each Excel formula with “=” in the query.

For example,

```
SELECT 45.5 AS PRODPRICE, 30 AS AMOUNT, '=A2*B2' AS PRICE
FROM SYSIBM.SYSDUMMY1
```

About this task

To save query results to a file:

Procedure

1. Open the **Export Query Results** wizard in one of the following ways:

- With a query result set opened in the editor, select **File > Export** to open the **Select** page of the **Export** wizard, and then select **Result set** from the list of available export wizards.
- With a query result set opened in the editor, click the **Export** toolbar button.

2. From the **Source** list, select the query result set that you want to export.

Note: If the query that you are exporting includes several result sets, all of them are displayed in the **Source** field. Also, if the query includes several result sets, all of them can be exported into one .xls or .xlsx file.

3. If you are using QMF for Workstation, specify a name for the file that you are creating in the **File name** field.

By default, the file is saved in your home directory.

4. Specify the type of file that you will be creating. Select from the list of available file types in the **Export type** field. You can choose one of the following file types:

- **PDF** (*.pdf)
- **CSV** (*.csv)
- **IXF** (*.ixf)
- **XML** QMF Markup Language file (*.xml)
- **HTML** (*.htm)
- **TEXT** (*.txt)
- **dbase III** (*.dbf)
- **XLS** Microsoft Excel 97-2003 (*.xls)
- **XLSX** Microsoft Excel 2007 (*.xlsx)

Note: Some of the fonts (for example, TTC) can be displayed incorrectly in PDF files.

5. If you selected any option different from **XLS** or **XLSX**, click **Next**. The Second page of the **Export Query Results** wizard opens. On this page you can select additional options that are available depending on the type of file that you are creating. For example, the wizard brings up a box that allows you to enter options for those grayed out choices when you chose DATAFORMAT=TEXT.

6. If you selected **IXF**, **dbase III**, or **XML**, after setting up the export options, click **Next** to go the page where you can set up the columns of the exported result set.

Note: Click **Finish** to skip the next step and close the **Export Query Results** wizard.

7. You can set up the columns by using following options:

- In the **Type** column, select a data type from the list.

- In the **Length** field, specify a maximum number of characters in a string.
 - In the **Precision** field, specify a maximum total number of digits for the DECIMAL type.
 - In the **Scale** field, specify a maximum number of fractional digits for the DECIMAL type.
8. Click **Finish**. If you are using QMF for Workstation, a save window specific to your operating system opens. If you are using QMF for WebSphere, a save window opens that lists the available files for download. Click a hyperlink to download the file to your workstation.

Specifying file export options

You must specify one or more of the following options when you export query results to a file. The options that you specify depend on the type of file that you are exporting.

| Field | File Type | Description |
|--------------------------------|--|--|
| What Data | <ul style="list-style-type: none"> • .txt • .html • .csv • .ixf • .wqml • .xml • .dbf | <p>You use the following radio buttons to specify what data in the query results will be saved:</p> <ul style="list-style-type: none"> • All to specify that all of the data in the current results grid will be saved with all the changes applied to it. For example, if you remove a column from the layout, it will not be saved in the export file. If your result set grid includes any calculated columns, they are included in the exported file along with other columns. • All - Result set only to specify that all of the data in the original query result set not depending on the changes applied to results grid will be saved. For example, if you remove a column from the layout, it will be saved in the export file anyway. • Selection to specify that only the data that you have selected in the current query results will be saved. This can include calculated columns if they are selected. The query result data that you want to save must be selected before you open the Export Query Results window. |
| Include Column Headings | <ul style="list-style-type: none"> • .txt • .html • .csv | Select this check box to include the query results column headings as the first row in the file. |

Table 20. Text options (continued)

| Field | File Type | Description |
|--|---|---|
| Date or Time Format | <ul style="list-style-type: none"> • .txt • .html • .csv | <p>Select one of the radio buttons to specify how the date and time should be formatted in the file:</p> <ul style="list-style-type: none"> • ISO to specify that the date and time will be formatted in the file based on the ISO format. The date will be formatted as (YYYY-MM-DD) and the time will be formatted as (HH:mm:ss). • USA to specify that the date and time will be formatted in the file based on the USA format. The date will be formatted as (MM/DD/YYYY) and the time will be formatted as (HH:mm PM/AM). • JIS to specify that the date and time will be formatted in the file based on the JIS format. The date will be formatted as (YYYY-MM-DD) and the time will be formatted as (HH:mm:ss). • EUR to specify that the date and time will be formatted in the file based on the EUR format. The date will be formatted as (DD.MM.YYYY) and the time will be formatted as (HH.mm.SS). • Custom to specify a custom format for date and time values. "Date": and "Time:" text fields accept Java format strings. (See http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html for more details.) |
| CCSID | <ul style="list-style-type: none"> • .txt • .html • .csv • .ixf • .dbf | <p>Use this field to select the code page to use when saving this file.</p> <p>Note: When changing the CCSID for exporting to an .ixf file in the PC/IXF Version 2 format, be aware that:</p> <ul style="list-style-type: none"> • Data lengths in columns in the exported file can change. • If the data length becomes greater after changing the encoding, the longer data will be truncated in the exported file. For example, truncation is probable when applying the UTF-8 encoding. |
| Display null values | <ul style="list-style-type: none"> • .txt | Use this field to specify the string that will be used to display null values. |
| Separate columns | <ul style="list-style-type: none"> • .txt | Use this field to specify the string that will be used to separate the columns. |
| Delimit values | <ul style="list-style-type: none"> • .txt | Use this field to specify the string that will be used to delimit values. |
| When should values be delimited | <ul style="list-style-type: none"> • .txt | <p>Use this field to specify when values should be delimited. Your choices are:</p> <ul style="list-style-type: none"> • Never • Only when necessary • Always |

Table 20. Text options (continued)

| Field | File Type | Description |
|---|--|---|
| Which type of values should be delimited | <ul style="list-style-type: none"> .txt | <p>Select one or more of the following check boxes to specify what types of values should be delimited in the file:</p> <ul style="list-style-type: none"> Column headings Character values Numeric values Date/time values |
| Format | .ixf | <p>Select one of the following radio buttons to specify how to format the .ixf output file. Your choices are:</p> <ul style="list-style-type: none"> PC/IXF (Version 1) to export data in PC/IXF Version 1 format. PC/IXF (Version 2) to export data in PC/IXF Version 2 format. <p>Note:</p> <ul style="list-style-type: none"> If you change the encoding for the exported file from the original encoding, data lengths in columns in the exported file can change. If the data length becomes greater after changing the encoding, the longer data will be truncated in the exported file. For example, truncation is probable when applying the UTF-8 encoding. <ul style="list-style-type: none"> System/370 IXF to export data in System/370 IXF format. |
| Add Document Information | .pdf | <p>Select this check box to add identification information in the document's properties fields when the PDF is created. Document information provides basic data about the document as well as criteria to facilitate searching. The document information includes:</p> <ul style="list-style-type: none"> Title specifies the title of the PDF document. Author specifies the author of the PDF document. Subject specifies the subject of the PDF document. Keywords Specifies the keywords for the PDF document. Keywords are used as search criteria. |
| Use only ANSI Fonts | .pdf | <p>Select this check box to specify that only ANSI fonts will be used in creating the PDF. If the PDF document includes only English characters, select this check box to produce a smaller PDF in less time.</p> <p>Note: Some of the fonts (for example, TTC) can be displayed incorrectly.</p> |

Importing query results from CSV and text files

You use the **CSV File Import** wizard to import CSV and text files to the data source.

About this task

To import a CSV or TXT file:

Procedure

1. Select **File > Import**.
The **Import** wizard opens.
2. Select **CSV File** and click **Next**.
The **Import result set from file** page of the **Import** wizard opens.
3. Click **Browse** next to the **File name** field.
Select either **CSV file (*.csv)** or **Text file (*.txt)** in the **Open** window to choose a CSV or TXT file to import.
4. Specify what encoding to use when importing the file from the **Encoding** list.
5. Click **Next**.
The **Select separator options** page opens.
6. Specify the separator options of the CSV or TXT file and click **Next**.
The **Result set structure** page opens.
7. Specify column options of the result set in the **Field definitions** table and click **Finish**.

Note: When you import a query result set from a CSV or TXT file, QMF scans the first 1000 rows of the result set to determine the maximum length of the VARCHAR data type. If the subsequent rows contain the VARCHAR data type that is longer than this maximum value, the cell value is truncated. You can increase the length of the data type in the **Field definitions** table.

The **CSV File Import** wizard closes and the new result set opens in the editor window.

Specifying separator options for CSV and TXT files

You use the **Select separator options** page of the **CSV File Import** wizard to specify how QMF will convert the data of the CSV or TXT file into the columns of the result set.

About this task

The separator options of the **CSV File Import** wizard allows you to specify how QMF interprets the format of the CSV or TXT file to convert it into a new result set, and how the finished result set will look. To specify separator options for the CSV or TXT file:

Procedure

1. Open the **Select separator options** page of the **CSV File Import** wizard.
2. Select the **Skip rows starting with** check box and specify a string in the text field to exclude certain rows from the result set.
3. Select what symbol is used to display decimal points in the file in the **Decimal separator** drop-down list.
4. Select the **First row contains column names** check box to specify that the first line of the file contains the names of columns to be created.
5. To specify that the columns will be created based on a fixed set of character lengths, click **Fixed width** and then specify the specific widths of the columns by entering the character lengths in the **Specify columns width** field.
When each character length is reached in the text, the characters after that point are placed in a separate column, regardless of the content of the text.
6. To specify that the columns will be created automatically by breaking at certain symbols, click **Separated by** and then selecting the symbols check boxes below.
When the result set is made, each time these symbols are encountered in the text, the characters after the symbol are placed in a separate column.
7. Select the **Merge separators** check box to consolidate columns that are created by multiple adjacent separators into one column.

For example, if **Comma** and **Space** are selected and the text includes the string NAME , DEPT, three columns will be created. The first column contains NAME, and was created when the comma was encountered. The second column is empty, and was created when the space was encountered. The third column contains DEPT, and was created when the remaining text was encountered. By selecting the **Merge separators** check box, only one column is created when the comma and the space are encountered.

8. Select whether a single or double quotation marks are used to delineate text in the **Text qualifier** drop-down list. Text that is contained in the specified quotes is always displayed as is, whether it contains a separator symbol or not.

Note: You can track how each selection changes the layout of the result set in the **Preview** table.

Specifying the result set structure of CSV and Text files

You use the **Result set structure** page of the **CSV File Import** wizard to customize the columns of the result set, and which columns are displayed.

About this task

By specifying the options in the **Field definitions** table, you can customize the structure of the imported result set:

Procedure

1. Select what columns to include in the result set by checking the check boxes to the left of the **Field name** column.
2. To edit a column name, click a cell in the **Field name** column and enter a new name.
3. To specify a different data type, click a cell in the **Type** column and select a data type from the list. Available options are the following:
 - **INTEGER** can be used to store signed whole numbers between -2 147 483 648 and 2 147 483 647.
 - **BIGINT** can be used to store large signed whole numbers between -9 223 372 036 854 775 808 and 9 223 372 036 854 775 807.
 - **DOUBLE** can be used to store large numbers with a floating decimal point.
 - **CHAR** is used to store fixed-length string of characters. For example, you can use it to store employee phone number which is generally fixed in length.
 - **VARCHAR** is used to store variable-length string of characters like employee name.
 - **DECIMAL** is used to store the fixed precision and scale numbers. This data type should be used for precise values, such as currency.
 - **DATE** is used to store a date only.
 - **TIME** is used to store a time only.
 - **TIMESTAMP** is used to store both a date and a time.

Note: DATE, TIME, and TIMESTAMP data types support different format options that define how the values are displayed in the result set.

Note: The supported ranges and the precision of values might vary depending on the database you use.
4. In the **Length** field, specify a maximum number of characters for the VARCHAR type.

Note: When you import a query result set from a CSV or TXT file, QMF scans the first 1000 rows of the result set to determine the maximum length of the VARCHAR data type. If the subsequent rows contain the VARCHAR data type that is longer than this maximum value, the cell value is truncated. You can increase the length of the data type in the **Field definitions** table.
5. In the **Precision** field, specify a maximum total number of digits for the DECIMAL type.
6. In the **Scale** field, specify a maximum number of:

- fractional digits for the DECIMAL type
 - digits in the fractional seconds part for the TIMESTAMP or TIMESTAMP WITH TIMEZONE type
7. In the **Preview** pane, you can track how each selection changes the layout of the result set.
 8. Click **Finish**.
The **CSV File Import** wizard closes and the new result set opens in the editor window.

Importing data contained in IXF files

Data contained in IXF files can be directly opened in the application.

Opening the contents of an IXF file for further processing

Once opened, the contents of the IXF file are presented in a results view, much the same as the output from a query.

About this task

The results view then allows you to format the data, generate a report, export it to a file or database table, offering options identical to those available when viewing query results.

Procedure

1. Click the **Open from file** toolbar button and enter the IXF file name.
The data contained in the IXF file opens as a results view in the editor.
2. Use the **Results** menu to run the following commands:
 - **Font** opens the **Font** page of the **Layout Properties** window where you can view and change the font attributes of your selection within the query results.
 - **Format** opens the **Format** page of the **Layout Properties** window where you can view and change the formatting options for the selected column in the query results.
 - **Grouping and Aggregation** specifies logical groupings of information within the query results.
 - **Show Total Summaries** shows summary information for the selected total columns.
 - **Add Calculated Column** opens the **Calculated Columns** window enabling you to add a calculated column to the query results. A calculated column is a column of data that is dynamically created based on an expression that you define.
 - **Reset Formatting** resets the formatting that has been applied to selected columns in the query results to the original formatting attributes.
 - **Reset All** resets the formatting that has been applied to all the columns in the query results to the original formatting attributes.
 - **Autofit** automatically fits the contents of columns and rows.
 - **Sort** opens the **Sort** window where you can set up multiple levels of sort criteria.
 - **Display Report** generates a visual or classic report using the query results that are active in the editor.
 - **Display Excel Sheet** opens Microsoft Excel and displays the query results data in an Excel data sheet.
 - **Export** opens the **Export Query Results** wizard where you can choose to save the query results to a file or a database.

Directly importing an IXF file into a database table

You can also directly import the contents of an IXF file to a database table.

About this task

To import data contained in IXF files into a database table:

Procedure

1. Select **File > Import**.

The **Import** wizard opens.

2. Select **Table** and click **Next**.

The **Import table data from file** page of the **Import Table** wizard opens.

3. Click **Browse** next to the **File name** field and enter the IXF file name.

4. Click **Next**.

The Select results and destination information page of the **Import Table** wizard opens.

5. Select the database from the **Database** list.

6. Specify the table owner name in the **Table owner** field and the table name in the **Table name** field.

7. Click **Next**.

The **Set up save options** page of the **Import Table** wizard opens.

8. Specify **All - Result set only** for the **What data?** option.

Since you are retrieving data from a file rather than a query, all rows and columns will be imported.

9. Specify the **Existing Data** option.

You use the following radio buttons to specify how the existing data will be saved to the database table:

- **Replace any existing data** to replace any existing data in the database table with the contents of the IXF file.
- **Add to any existing data** to append the IXF file contents to any data that already resides in the database table.

10. Click **Finish**.

The **Import Table** wizard closes and the data contained in the IXF file is saved to the database table.

Importing query results from Excel files

You can import the content of an Excel file to the data source.

About this task

To import an Excel file:

Procedure

1. From the main menu, select **File > Import**.

The **Import** wizard opens.

2. Select **Excel File** and click **Next**.

The **Excel File Import** page of the **Import** wizard opens.

3. To specify the file that you want to import:

- a) Click **Browse** next to the **File name** field.
- b) Select either ***.xls** or ***.xlsx** in the **Open** window.

4. If the document that you want to import includes several Excel sheets, select one of the following options:

One query with several result sets

Select this option if you want to create a single query with multiple result sets from your Excel file.

Several queries with one result set

Select this option if you want to create a query with a single result set from each sheet in your Excel file.

5. In the **Sheets** area, select the sheets that you want to import and click **Next** to specify the import options for each sheet.
6. From the **Sheet** list, select the sheet of the Excel file you want to work with.
Note: All import settings that you specify for the first Excel sheet are automatically applied to every other sheet that you are importing. You can then change the settings for each individual sheet.
7. In the **First row number** field, type the row number of the Excel sheet that you want to be the first row in the result set.
8. In the **First column number** field, type the column number of the Excel sheet that you want to be the first column in the result set.
9. To limit the number of rows in the result set, select the **Row limit** check box and type the number of rows.
10. To limit the number of columns in the result set, select the **Column limit** check box and type the number of columns.
11. If you want to turn the first row of the Excel document into the column names of the result set, select the **First row contains column names** check box.
12. In the **Preview** pane, you can track how each selection changes the layout of the result set.
13. Optional: Repeat steps [6-12](#) for each sheet.
14. Click **Next** to specify the structure of the table from the imported Excel file.
15. From the **Sheet** list, select the sheet of the Excel file you want to work with.
16. In the **Field definitions** area, select what columns to include in the result set by checking the check boxes to the left of the **Field name** column.
17. To edit a column name, click a cell in the **Field name** column and enter a new name.
18. To specify a different data type, click a cell in the **Type** column and select a data type from the list. Available options are the following:
 - **INTEGER** can be used to store signed whole numbers between -2 147 483 648 and 2 147 483 647.
 - **BIGINT** can be used to store large signed whole numbers between -9 223 372 036 854 775 808 and 9 223 372 036 854 775 807.
 - **DOUBLE** can be used to store large numbers with a floating decimal point.
 - **VARCHAR** is used to store variable-length string of characters like employee name.
 - **BIT** is an integer data type. Possible values are 1, 0, or NULL. The string value **TRUE** is converted to 1, empty strings are converted to NULL and other string values are converted to 0. Converting any numerical values to the BIT data type promotes any nonzero value to 1.
 - **DECIMAL** is used to store the fixed precision and scale numbers. This data type should be used for precise values, such as currency.
 - **DATE** is used to store a date only.
 - **TIME** is used to store a time only.
 - **TIMESTAMP** is used to store both a date and a time.**Note:** DATE, TIME, and TIMESTAMP data types support different format options that define how the values are displayed in the result set.
Note: The supported ranges and the precision of values might vary depending on the database you use.
19. In the **Length** field, specify a maximum number of characters for the VARCHAR type.

20. In the **Precision** field, specify a maximum total number of digits for the DECIMAL type.
21. In the **Scale** field, specify a maximum number of:
 - fractional digits for the DECIMAL type
 - digits in the fractional seconds part for the TIMESTAMP or TIMESTAMP WITH TIMEZONE type
22. Optional: Repeat steps [15-21](#) for each sheet.
23. Click **Finish**.

Results

The imported data is displayed on the screen in the form of one or several query result sets. For detailed information on working with query result sets, see the *Managing query results* topic.

Importing table contents of a saved file into a database table

You can import the table contents that are saved in a file into a database table.

About this task

To import data contained in a file into a database table:

Procedure

1. Select **File > Import**.
2. From the list of available wizards select **Table** and click **Next**.
The **Import table data from file** page of the Import Table wizard opens.
3. Click **Browse** next to the **File name** field and specify the file name and location.

The following file formats are supported:

- *.tab
- *.ctab
- *.txt
- *.xml
- *.ixf
- *.csv
- *.dbf
- *.xls
- *.xlsx

4. Click **Next**.
 - If you import the table contents from *.txt or *.csv files, the **CSV File Import** page opens. For information about the import options, see [“Importing query results from CSV and text files” on page 203](#).
 - If you import *.xls or *.xlsx files, the **Excel File Import** page opens. For information about the import options, see [“Importing query results from Excel files” on page 207](#).

Note: If the Excel file that you want to import includes multiple spreadsheets and you want to import at least two of them, you must access the **Excel File Import** wizard in a different way. From the main menu, select **File > Import**. In the **Import** wizard, select **Excel file** and click **Next**.

5. Click **Next**.
The **Select results and destination information** page opens.
6. Select the database from the **Database** list.
7. Specify the table owner name in the **Table owner** field and the table name in the **Table name** field.

8. Click **Next**

The **Set up save options** page of the **Import Table** wizard opens.

9. Specify **All - Result set only** for the **What data?** option.

Note: Since you are retrieving data from a file rather than a query, all rows and columns are imported.

10. Specify the Existing Data option.

Use the radio buttons to specify how to save the existing data to the database table:

- **Replace any existing data** to replace any existing data in the database table with the contents of the import file.
- **Add to any existing data** to append the import file contents to any data that already resides in the database table.

11. Click **Finish**.

Results

The **Import Table wizard** closes and the data contained in the file is saved to the database table.

Saving data objects as dynamarts

Dynamart is a format for saving result sets with all the modifications.

About this task

When you save a data file as a dynamart, the link to this file is saved within it. When you open the dynamart, it reloads the data from the linked source file and updates the result set if the source file was modified or replaced. You can change the source object or deactivate the data update for a particular dynamart in the **Dynamart Settings** window.

When you save a query as a dynamart, the result set and the operations with it are saved within the dynamart. When you open the dynamart, it reloads the result set and automatically applies all the custom changes, like filtering or adding calculated columns.

To save data objects as dynamarts:

Procedure

1. Make sure that the data file that you want to save as a dynamart is opened in the active window. If you work with an analytical or visual query, run it before you save it as a dynamart.
2. From the main menu, select **File > Save As**.
3. Select one of the following options depending on where you want to save the dynamart:
 - **Save to Repository**
 - **Save to File**
4. In the **Save to Repository** window, specify the parent entry, file name, and comment.
5. In the **Save to File** window, specify the directory to save the file and the file name. You can click **Browse** to search the destination directory.
6. Select one of the following options from the **Type** list:
 - **Dynamart** to save a visual query, CSV, DBF, IXF, TXT, or XML files.
 - **Compound Dynamart** to save an analytical query.
7. Click **Finish** to save the dynamart files.

Managing source data files in dynamarts

After saving the dynamart, you can change or completely remove the link to the source data object.

About this task

When you save data files as dynamarts, the link to this data file and the time of its last modification are saved within the dynamart. When you open this dynamart, QMF reloads the data from the linked source file and updates the result set if the source file was modified or replaced.

You can change the source object or disable data update for a particular dynamart in the **Dynamart Settings** window.

To replace the source data file or disable data update:

Procedure

1. Select **Results > Dynamart Settings**.
2. In the **Dynamart Settings** window, specify the name and location of a data file that you want to use as source file in the **Data file** field. You can click **Browse** to search the file.
3. Select the **Update dynamart when file is changed** check box if you want the result set to be updated on any changes in the linked data file.
4. Clear the **Update dynamart when file is changed** check box to unlink the source data file. In this case the dynamart always displays original data even if the data file is replaced or updated.
5. Click **OK** to close the **Dynamart Settings** window.

Related tasks

[“Setting root directory location for dynamart source files” on page 17](#)

You can specify a root directory location where dynamart source data files are stored.

Working with LOB data in the query results

When retrieving large object data you must identify the application that will be used to view the LOB data that has been included in the query results column.

About this task

Retrieving and saving large amounts of data can be time consuming. In order to prevent excessive resource consumption, certain restrictions are placed on a user's ability to retrieve and save large object data. Your ability to retrieve and save LOB data is dictated by the LOB Option resource limit set for your user ID or group by your QMF administrator.

To retrieve or save LOB data:

Procedure

1. Run a query or procedure that calls the table where the LOB data is stored on the database.
2. The LOB data is retrieved from the data source. The contents of each cell in the column that contains LOB data can appear differently depending on the LOB resource limit that has been set for your user ID:
 - If the **LOB Retrieval Option** resource limit is **Retrieve LOB data automatically**, then all the LOB data is available to you and displayed in the query results.

This option pulls ALL of the LOB data for ALL of the LOB columns from the database to the local computer. The actual LOB data is not displayed in grids and reports. Instead, pointers to the LOB data are displayed.

Note: This option can potentially consume a large amount of resources on the local computer.

- If the **LOB Retrieval Option** resource limit is **Retrieve LOB data on demand**, then each cell in the column containing LOB data is identified with the label, `LOB_locator`. You must initiate the retrieval of the of the LOB data.
3. To initiate the retrieval of the LOB data, double-click the cell labeled `LOB_locator`. The **LOB Type Associations** window opens. You can also open the **LOB Type Associations** window by right-clicking in the header cell of the query results column that contains the LOB data and selecting **LOB Type Associations** from the pop-up menu.
 4. From the **LOB Type Associations** window you will identify the type of LOB data that is included in each of the cells of the query results column. The application that will be used to open the LOB data file will be determined based on the file extension that you select. You can select one of the following:
 - Select **File Extension** to specify the type of LOB data file that will be included in each cell of the query results column. Choose from the drop-down list of available extensions. This option will apply the extension that you select to all cells in the query result column.
 - Select **Mapping Column** to individually specify which application will be mapped to each LOB data cell in the query results column. From the drop-down list, select the column in the query results that contains the file extension information for the LOB data. Click **Mapping**. The **File Extension Mapping** window opens.
 - a. The **Mapping table** includes the following fields:
 - **Mapping Value**: Lists each unique data type that exists in the mapping column.
 - **File Extension**: Contains the file extension of the unique data type, and the application that will be used to open the LOB data.
 - b. Click the **File Extension** field. Type the file extension you want to represent the unique data type. For example, if the mapping column in the query results contains the text string "bitmap", you would type a file extension of .bmp. You can also select from a drop-down list of pre-defined file extensions and their associated applications that will be used to open that type of file.
 - c. When all file extensions have been specified, click **OK**. Control returns to the **LOB Type Associations** window.
 5. Click **OK**. The **Column LOB Type Associations** window closes.
 6. To retrieve and view the LOB data you double-click LOB data's cell. The LOB data opens in the designated application. In the query results grid, the label for the cell will change from `LOB_locator` to `LOB_value` to indicate that the LOB data has been retrieved. You can Select **Retrieve all LOB** values to retrieve all large object data files that are included in the query results column.

Overriding LOB resource limit option

If the QMF administrator has granted your user ID permission, you can override the LOB resource limits that have been set for your resource limits group.

About this task

To override the LOB retrieval resource limit:

Procedure

1. Select **View > Preferences**. The **Preferences** window.
2. Click **LOBs** to change the default options for retrieving LOBs.
3. Check the **Override LOB options if possible** check box to make changes to the LOB options. If this check box is not available, your user ID does not have permission to override LOB resource limits.

Viewing and editing the contents of a cell in the query results

If you are using QMF for Workstation, you can use the **Zoom** option to view in hexadecimal or binary format the contents of a cell in the query results.

About this task

This topic describes how to:

- View the contents of a query results cell when the results are read-only. See [step 1](#).
- View and edit the contents of a query results cell when the results are writable (not read-only). See [step 2](#).

Procedure

1. To view the contents of a query results cell when the results are read-only:
 - a) Right-click a cell in the query results. Select **Zoom** from the pop-up menu. The **Zoom** window opens.
 - b) The **CCSID** field specifies the code page that was used to save the data in the selected query results cell.
 - c) The current value of the selected query results cell is displayed in the **Cell value** field.
 - d) Click **Advanced** to expand the dialog with additional fields to display the data in hexadecimal or binary format. This button is disabled for cells with numeric data.
 - e) Select **Hexadecimal** to display the data in hexadecimal format. Select **Binary** to display the data in binary format. The **Zoom** window displays samples of the data in the format that you have chosen.
 - f) Select **Cancel** to close the Edit window.
2. To view or edit the contents of a query results cell when the results are writable (not read-only):
 - a) Right-click a cell in the query results and select **Edit** from the pop-up menu.

The **Zoom Edit** window opens.

The **CCSID** field specifies the code page used to save the data in the selected query results cell.

The current value of the selected query results cell is displayed in the **Cell value** field. The **Cell value** field is editable.
 - b) Make the necessary changes and select **OK** to apply them.
 - c) Click **Advanced** to expand the dialog with additional fields to display the data in hexadecimal or binary format.

This button is disabled for cells with numeric data.
 - d) Select **Hexadecimal** to display the data in hexadecimal format or select **Binary** to display the data in binary format.

The **Zoom Edit** window displays samples of the data in the format that you have chosen.
 - e) Select **Cancel** to close the **Zoom Edit** window.

To edit a cell in the tables on which the query is based you can use the table editor.

Viewing multiple result sets

There are two instances where multiple query result sets might be returned. The first instance is as a result of running a stored procedure. The second instance is as a result of running a single query that has multiple SQL statements.

About this task

All result sets are returned to the editor window and assigned an index number starting with the number one (1). Each set of query results is identified in a drop-down list on the query editor toolbar. To view a specific result set, select it from the drop-down list.

Results menu

The **Results** menu becomes available when the results of a query are returned and are the active contents in the editor.

The following menu commands are available:

Retrieve All

Select **Retrieve All** to finish retrieving all the query results data. When retrieving large amounts of data, unless otherwise specified, only the first 100 data rows are retrieved.

Font

Opens the **Font** page of the **Layout Properties** window where you can view and change the font attributes of your selection within the query results.

Format

Opens the **Format** page of the **Layout Properties** window where you can view and change the formatting options for the selected column in the query results.

Grouping and Aggregation

Use the Grouping and Aggregation menu choices to specify logical groupings of information within the query results. The following options are available:

- Select **No Aggregation** to remove any aggregation or summary formation from the selected column.
- Select **Top Group** to group the selected column without summary information in the top of the query results.
- Select **Side Group** to group the selected column without summary information to the left side of the query results.
- Select **Count** to summarize the total number of values in the column.
- Select **First** to create a summary row showing the first value in the column.
- Select **Last** to create a summary row showing the last value in the column.
- Select **Maximum** to create a summary row showing the maximum value in the column.
- Select **Minimum** to create a summary row showing the minimum value in the column.
- Select **Sum** to create a summary row showing the sum of the values in the column.
- Select **Cumulative Sum** to display the values in the column as a cumulative sum, with a summary.
- Select **Average** to create a summary row showing the average of the values in the column.
- Select **Standard Deviation** to create a summary row showing the standard deviation of the values in the column.
- Select **Percentage of Group** to display the values in the column as a percentage value of the group total, with a summary row showing the sum of the percentages.
- Select **Percentage of Total** to display the values in the column as a percentage value of the column total, with a summary row showing the sum of the percentages.
- Select **Cumulative Percentage of Group** to display the values in the column as a cumulative percentage value of the group total, with a summary row showing the sum of the percentages.

- Select **Cumulative Percentage of Total** to display the values in the column as a cumulative percentage value of the column total, with a summary row showing the sum of the percentages.

Show Summaries

Show summary information for the selected Top or Side group column.

Show Total Summaries

Show summary information for the selected total columns.

Drill Down

Expand the Top or Side group detail information.

Drill Up

Collapse the Top or Side group detail information.

Add Calculated Column

Opens the **Calculated Columns** window enabling you to add a calculated column to the query results. A calculated column is a column of data that is dynamically created based on an expression that you define.

Reset Formatting

Resets the formatting that has been applied to selected columns in the query results to the original formatting attributes.

Reset All

Resets the formatting that has been applied to all the columns in the query results to the original formatting attributes.

AutoFit

Select **AutoFit** to automatically fit the contents of one of the following:

- **Selected Column(s)** to automatically fit the contents of the selected columns into the same column size.
- **All Columns** to automatically fit the contents of all the columns in the query results into the same column size.
- **All Rows** to automatically fit the contents of all the rows in the query results into the same row size.
- **All** to automatically fit the contents of the entire query results.

Sort

Opens the **Sort** window where you can set up multiple levels of sort criteria.

Go to

Opens the **Go to Row** window, where you can navigate to a specific row.

Find

Initiates a search of the query results for a specific string. You specify the search string using the **Find** window.

Display Report

Generates visual or classic report using the query results that are active in the editor. You can choose to generate the report based on the existing formatting of the query results or simpler default formatting.

Display Excel Sheet

Opens Microsoft Excel and displays the query results data in an Excel data sheet. (QMF for Workstation only). If you are exporting a query with several result sets, each of the result sets will be displayed in a separate workbook.

Note: For relational query results whose columns have both breaks and usage codes, an Excel PivotTable will be created automatically. Processing time for this functionality increases depending on the amount of data.

Retrieve All LOB Values

Retrieves all LOB values in the query results set.

Export

Opens the **Export Query Results** wizard where you can choose to save the query results to a file or a database.

Display Chart

Generates a chart from a query result set that is open in the editor.

Filter Results

Opens the **Dynamart Filter** wizard where you can create filters that control the contents of the query result set. For more information about filtering query result sets, see [“Filtering query results by using the Dynamart Filter wizard” on page 187.](#)

Sorting query results

Use the **Sort Columns** dialog to sort the query results on columns of data.

About this task

Perform the following steps to specify the columns by which the query results are sorted:

Procedure

1. With query results in the available window, select **Results > Sort**.
The **Sort Columns** dialog box is displayed.
2. Select the sort orders for the desired columns in the **Sort Order** fields. Query results are sorted using the column whose sort order is set to 1 first and then based on the column whose sort order is set to 2 and so on.
Note: You can limit the number of columns that can be used to sort the query result using the DSQCP_SORT_COL_SUPPORTED global variable. You can set a desired value in this global variable or set 0 to allow sort on all columns.
3. Specify the sort sequence, **Ascending** or **Descending**, for the desired columns in the **Sort Sequence** fields.
4. Select the sort type for the desired columns in the **Sort Type** fields.
5. By default, the **Sort string based on locale rules** check box is not selected indicating that the query results are sorted based on the comparison of characters according to their Unicode code point values. Select the **Sort string based on locale rules** check box to sort query results based on a character's position in the national language alphabet that has been defined for the current locale.
6. Click **OK**.

The query results are sorted according to the conditions specified in the **Sort Columns** dialog box.

The specified sorting order is saved within the query and used the next time you run it if the DSQQW_QUERY_PRESERVE_SORT global variable is set preserve sorting order (1). If you need to rerun the query without applying the specified sorting order, set the DSQQW_QUERY_PRESERVE_SORT global variable to zero (do not preserve sort order), and then run the query. You will always be able to apply the specified sorting order from the **Sort Columns** dialog box.

Specifying print options for query result sets and quick reports

In QMF for Workstation, you can specify the page setup options that will be used when you print the visual query or quick report opened in the editor window.

About this task

To control the appearance of printed pages for query result sets or quick reports:

Procedure

1. With an active object in the editor window, open the **Page Setup** window by selecting **File > Page Setup**.
2. Specify the printing options that will apply to your object.

The sets of options vary depending on the type of object you want to print. The following table shows what page options you can specify for query result sets and quick reports:

| Query result set | Quick report |
|---|---|
| Printer and page parameters including the type and size, orientation and margins | Paper parameters including the type and size, orientation and margins |
| Header and Footer text | Footer text |
| Whether to print row and column numbers, column headers, grid lines, aggregation type, and page header and footer | Whether to print row and column numbers, column headers and page footer |
| Page order | Fixed groups |
| Fixed side and top groups | |

Note: You can use special codes to insert specific data to header and footer fields.

3. To make the specified values default, click **Set as Default**.
4. Click **OK** to close the **Page Setup** window. The options that you specified are used to control the appearance of the printed object.

Codes to insert specific data in a page header and footer

You can use special codes to insert specific data in a header or footer.

When you print a visual query result set or a quick report, you might want to add some data like the document name or the total number of printed pages to the page header or footer. To do this, you can use special codes that are described in the following table:

| Code | Description |
|-------------|--|
| &f | Prints the name of the object. |
| &n | Prints the current page number. |
| &N | Prints the total number of pages. |
| &d | Prints the current date. |
| &t | Prints the current time. |
| &< | Prints the number of the first printed page. |
| &> | Prints the number of the last printed page. |
| && | Prints a single ampersand. |
| & | Defines the border of the left, central and right parts of a header or footer. |

Examples

| | |
|-------------|---|
| & &n | The current page number is printed in the center of a header (footer). |
| & & &d | The current date is printed on the right side of the header (footer). |
| &f & & &d | The object name is printed on the left side of the header (footer); and the current date is printed on the right side of the header (footer). |

Saving data using the regular and fast save methods

When you are saving query results data to a database, you can choose to save the data using a "regular save" or a "fast save" method.

When saving your query results data using a "regular save" method, the interface that you are using (QMF for Workstation or QMF for WebSphere) saves the retrieved query results to the database using an individual SQL INSERT statement for each row of data. This type of save can be expensive in terms of performance due to the overhead of passing large amounts of data back to the database.

When saving your query results data using a "fast save" method, all the processing occurs at the database. There is no further passing of data between the QMF interface and the database. The QMF interface adds SQL to the original query, the query is rerun and the data is saved directly into the specified table(s). Saving query results data using the "fast save" process significantly improves performance when a large amount of data is to be saved.

Selecting the fast save method

If you elect to save your query results data using a "fast save" method, all ORDER BY clauses remain in the query that is run at the database. Keeping the ORDER BY clause retains the row order of the query results data.

```
Select [Columns] from [SourceTable]
Order By [Column]
```

Setting the regular or fast save method

To set the regular or fast save method:

1. With query results in the editor window, select **Export** from the **Results** menu. The **Export Query Results** window opens. Select **Database** from the **Export Query Results** window. Enter values in the **Export destination** fields. Click **Next**. The **Set up save options** page opens.
2. Select one of the following:
 - Click **Regular (send retrieved data back to data source and save using an insert statement for each row)** to select the regular save method.
 - Click **Fast (directly save results at the data source by re-running the query at the data source)** to select the fast save method that retains any ORDER BY clause(s).
 - Select the **Remove ORDER BY clause(s) from original SQL before re-running query** check box to select the fast save method that runs the query without the ORDER BY clauses. Use this option when the removal of ORDER BY clause(s) does not influence the returned data.

Selecting the fast save method with the remove ORDER BY clause option

If you elect to save your query results data using the "fast save" method, you can choose to have the QMF interface remove all ORDER BY clauses from the query before the query is sent to the database.

In keeping with the example used in [“Selecting the fast save method”](#) on page 218, the following query would be sent:

```
Insert into [DestTable]
Select [Columns] from [SourceTable]
```

In most cases, removing the ORDER BY clauses has no effect on the query result data that is saved. In some cases, however, removing the ORDER BY clause from the query can affect the result set that is saved. This will happen if the original query limits the number of rows returned. For example, removing the ORDER BY clause from the following query changes the results:

```
select x,y,z from table
order by x
fetch first 10 rows only
```

Setting the regular or fast save method for procedures

You can set up a regular or fast save method for your procedure query results data using the global variable DSQQW_FST_SV_DATA. The default value for DSQQW_FST_SV_DATA is the value zero (0).

1. Select **View > Preferences**. Expand the QMF folder. Select **Global Variables**. The **Global variables** window opens.
2. From the **Global variables** window select the DSQQW_FST_SV_DATA variable.
3. Select or specify:
 - 0 - Regular save
 - 1 - Fast save with ORDER BY clause(s) stripped
 - 2 - Fast save with ORDER BY clause(s)

You can also set the global variable in either of the following ways:

- Use the Set Global procedure command using either QMF interface. For example SET GLOBAL (DSQQW_FST_SV_DATA = 2).
- You can also set DSQQW_FST_SV_DATA to 2 using SetGlobalVariables() in the API for either QMF interface.

Transforming the Results tab into the Results editing view


You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

About this task

You can detach the **Results** tab for the following objects:


- Analytical Query
- DBF, CSV, IXF, or XML file
- Dynamart
- Forecast
- Visual Query

To transform the **Results** tab into the view in three ways:

1. From the main menu, select **Window > Show View > Other > General > Results**.
2. From the main menu, select **Query > Detach Results View** or **Forecast > Detach Results View**.
3. Click the **Detach Results View** () toolbar button.

The **Results** view becomes available in a separate view.

To transform the **Results** view into the tab:

1. Close the **Results** view.
2. From the main menu, select **Query > Detach Results View** or **Forecast > Detach Results View**.
3. Click the **Detach Results View** () toolbar button.

The **Results** view transforms into the tab.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Chapter 10. Working with scheduled tasks

Scheduled tasks allow you to create automated actions such as running queries and exporting query results without taking direct action from the user interface.

About this task

Scheduled tasks are invaluable tools that help to automate business processes to run at specified dates and times. This is especially useful for actions that must be run on a regular basis, such as weekly sales reports that are supposed to be distributed to several different locations, or quarterly earnings reports.

There are two ways of scheduling the tasks.

1. Procedure 1: Schedules tasks from the **File** menu.
2. Procedure 2: Schedules tasks from the **Repository Explorer > Workspace(s)** tree, by right clicking on the objects under the workspaces.

Procedure 1

To create a scheduled task, perform the following steps:

1. Select **View > Scheduled Tasks** or **File > New > Scheduled Task**.
2. In the **Task List** window, click **Select Repository Connection** and use the **Select Repository Connection** dialog to select the repository connection that you want the task to access.
3. If authorization information is required, you are prompted to enter it in the **Log on to** window.
4. Click **Add Task** to create a scheduled task if you use **Task List** dialog.
5. If you open the **Create New Scheduled Task** wizard by clicking **File > New > Scheduled Task**, select the type of scheduler you want to use. You can use a local scheduler or QMF for WebSphere scheduler and click **Next**.
6. If you open the **Create New Scheduled Task** wizard by right clicking on a QMF object such as procedure, query, visual report, and quick reports, then a new drop down is displayed with the User Selection. The User Selection is populated with the values you have selected in the current workspace context.
7. Specify a name for the task in the **Task name** field.
8. Click **Select Repository Connection** and use the **Select Repository Connection** dialog to select the repository connection that you want the task to access.
9. Click **Set User Information**.
The **Set User Information for Repository** window opens.
10. Specify the login name and password for the repository in the **Login** and **Password** fields and click **OK**.
The **Set User Information for Repository** window closes.
11. Click **Add Action**.
The **Action Wizard** opens.
12. Specify a name for the action in the **Action name** field.
13. Select what type of action you want to add and click **Next**.
14. Specify the options for the selected task type and click **Finish**.
The **Action Wizard** closes and the new action is added to the **Actions** list.
15. To generate a detailed log of this task execution, select the **Enable logging** check box. The error log and the task execution report are saved to the user home directory if the task is run from the operating system scheduler.

16. When you have added all of the actions that you want the task to run, click OK. The Task window closes and control returns to the Task List window.
17. Click **Import task** to import a task.
18. Click **Export task** to export a task.
19. To schedule a task, select the task that you want to schedule and click **Schedule task**.
20. Click **OK**.

The **Task List** window closes.

Procedure 2

Alternately, you can schedule a task by right clicking on one or multiple objects from the Repository explorer.

To create a scheduled task, perform the following steps:

1. Select one or more object(s) of type Procedure, Visual Query, Quick Report, Visual Report, from the **Repository Explorer** tree and right click on the selected objects. Click **New > Scheduled Task**.
2. From the **Create New Scheduled Task** window, on the **Select the task type** page, select the task type and click **Next**.

Note: If you are connected to a shared repository, you would be able to see only local scheduler option. If you are connected to a WebService repository connection, you will be able to see the following two options:

- Use local scheduler
 - Use QMF for WebSphere scheduler.
3. On a **Local task** or **Remote task** page (based on the type of the scheduler), enter the following:
 - **Task name:** Specify the name of the task.
 - **Repository connection:** By default, the current repository connection is selected. If you want to change the repository connection, select from the drop-down and provide user name and password.
 - **User Selection(s):** Select the object to be scheduled from the drop-down. You can even add a new object to the existing list in the drop-down by clicking on the **Select New Object** button.
 4. Click **Add Action**.

The Action Wizard opens.

5. Specify a name for the action in the **Action name** field.
6. Select the type of action from the following and click **Next**.
 - **Execute object:** The execute object action allows you to automatically execute queries, a visual report, procedure, or quick report, and export, mail, or print it. For more information on how to add execute object action, see [Adding an execute object action](#) .
 - **Resource manipulation:** The resource manipulation action allows you to automatically copy, paste, delete, and email resources that are generated by the scheduled task. For more information on how to add resource manipulation action, see [“Adding a resource manipulation task” on page 233](#).
 - **Action group:** You can use action groups to create a series of actions that should be performed successively. For more information on creating a series of action groups, see [“Adding an action group” on page 236](#).
7. Specify the options for the selected task type and click **Finish**.

The **Action Wizard** closes and the new action is added to the **Actions** list.

8. To generate a detailed log of this task execution, select the **Enable logging** check box. The error log and the task execution report are saved to the user home directory if the task is run from the operating system scheduler.

9. When you have added all of the actions that you want the task to run, click **OK**. The Task window closes and control returns to the Task List window.
10. Click **Import task** to import a task.
11. Click **Export task** to export a task.
12. To schedule a task, select the task that you want to schedule and click **Schedule task**.
13. Click **OK**.

The **Task List** window closes.

Adding an execute object action

The execute object action allows you to automatically execute queries, a visual report, procedure, or quick report, and export, mail, or print it.

About this task

To add an execute object action to a scheduled task:

Procedure 1

1. Open a task in the **Task List** window.
2. In the **Scheduled Tasks** view, select the task that you want to work with and click **Edit Task**.
3. Click **Add Action**.

The **Action Wizard** opens.

4. Specify a name for the action in the **Action name** field, click **Execute object**, and then click **Next**.

The **Specify object** page opens.

5. Select the type of object that you want to execute.
6. Specify the location of the object.

There are two options:

- To specify the static source location of the object, click **Path** and then click the ellipsis (...) at the bottom right of the **Source** area.

The **Open** window opens.

- Navigate to and select the object and click **Finish**.

The **Open** window closes.

- To specify a conditional source location for the object, click **Formula** and then click the ellipsis (...).

The **Formula editor** opens.

- Specify the formula to use to determine the location of the object and click **OK**.

The **Formula editor** closes.

7. Select **Set a new data source** check box to specify a particular data source against which you want to execute the object. Then click the ellipsis (...), the **Add Data Source** window opens, select the required data source and click **OK**.
8. To add an available data source, click **Add Data Source**. The **Add Data Source** window opens.
9. Navigate to and select the data source that you want to add, specify a login name and password for the data source, and click **OK**. The **Add Data Source** window closes.
10. Click **Next**.

If the object that you specified in [step 6](#) includes a prompt, the **Specify variables** page opens. Otherwise, the **Specify operations** page opens, and you must go to [step 11](#).

The **Edit Variable** window opens.

11. On the **Specify variables** page, double-click the variable that you want to work with.
12. In the **Edit Variable** window, specify the value for the variable or use the Formula Editor to create an expression.
13. Click **OK** to close the **Edit Variable** window, then click **Next**.

The **Specify operations** page opens.

14. On the **Specify operations** page, click **Add Operation**.

The **Operation Wizard** opens.

15. Select an operation and click **Next**.
16. Specify the options for the selected operation and click **Finish**.

The **Operation Wizard** closes.

17. When you have finished adding all of the operations that you want the action to run, click **Finish**.

The **Action Wizard** closes and control returns to the **Task** window.

18. Click **OK**.

The **Task** window closes and control returns to the **Task List** window.

19. Click **OK**.

The **Task List** window closes.

Procedure 2

If you have scheduled the task from the context menu, follow this procedure.

1. On the **Local Task** or **Remote Task** page, click **Add Action**.

The **Action Wizard** opens.

2. Specify a name for the action in the **Action name** field.
3. Select **Execute object** and click **Next**.

The **Specify object details** page opens.

4. By default, the location of the object is displayed in the **Path** field.

If you want to change the object, perform the following steps:

- **Path:**

- a. To specify the static source location of the object, click **Path** and then click the ellipsis (...) at the bottom right of the **Source** area.

The **Open** window opens.

- b. Navigate to and select the object and click **Finish**.

The **Open** window closes.

- **Formula:**

- a. To specify a conditional source location for the object, click **Formula** and then click the ellipsis (...).

The **Formula editor** opens.

- b. Specify the formula to use to determine the location of the object and click **OK**.

The **Formula editor** closes.

By default, the selected object's datasource details will appear in the **Data Sources** table.

5. Select **Set a new data source** check box to specify a particular data source against which you want to execute the object. Then click the ellipsis (...), the **Add Data Source** window opens, select the required data source and click **OK**.

6. To add an available data source, click **Add Data Source**. The Add Data Source window opens.
7. Navigate to and select the data source that you want to add, specify a login name and password for the data source, and click **OK**. The Add Data Source window closes.
8. Click **Next**.

If the object that you specified in [step 4](#) includes a prompt, the **Specify variables** page opens. Otherwise, the **Specify operations** page opens, and you must go to [step 9](#).

The **Edit Variable** window opens.

9. On the **Specify variables** page, double-click the variable that you want to work with.
10. In the **Edit Variable** window, specify the value for the variable or use the Formula Editor to create an expression.
11. Click **OK** to close the **Edit Variable** window, then click **Next**.

The **Specify operations** page opens.

12. On the **Specify operations** page, click **Add Operation**.

The **Operation Wizard** opens.

13. Select an operation from the following and click **Next**.

- **Export query:** You use the export query operation to export query results to a file. For more information, see [“Specifying export query operation options” on page 226](#).
- **Export query to mail:** You use the export query to mail operation to send query results in an email. For more information, see [“Specifying export query to mail operation options” on page 227](#).
- **Print query:** You use the print query operation to print the specified query results. For more information, see [“Specifying print query operation options” on page 229](#).

14. Specify the options for the selected operation and click **Finish**.

The **Operation Wizard** closes.

15. When you have finished adding all of the operations that you want the action to run, click **Finish**.

The **Action Wizard** closes and control returns to the **Task** window.

16. Click **OK**.

The **Task** window closes and control returns to the **Task List** window.

17. Click **OK**.

The **Task List** window closes.

Related tasks

[Adding a resource manipulation task](#)

The resource manipulation action allows you to automatically copy, paste, delete, and email resources that are generated by the scheduled task.

[Adding an action group](#)

You can use action groups to create a series of actions that should be performed successively.

[Specifying variables for actions and action groups](#)

You can create a scheduled task that runs a variety of data objects such as queries, quick reports, visual reports, and procedures containing parameters.

[Scheduling local tasks](#)

Use the specific to your operating system scheduler window or application schedule task window to schedule a task to run.

[Sending HTML tables in email](#)

You can send query results as HTML tables in the email when you work with scheduled tasks.

Specifying export query operation options

You use the export query operation to export query results to a file.

About this task

To add an export query operation to an action:

Procedure

1. Open the **Specify operations** page of the **Action Wizard** and click **Add Operation**.
The **Operation Wizard** opens.
2. Click **Export query** and then click **Next**.
The **Specify export parameters** page opens.
3. If the query has multiple result sets, select the result set that you want to export from the **Result set** list.
4. If the query has multiple display modes, select the display mode that you want to export from the **Display mode** list.
5. Select the file type that you want to export from the **Export type** drop-down list.

Valid file types are:

- **Text** (*.txt)
 - **CSV** (*.csv)
 - **PDF** (*.pdf)
 - **HTML** (*.html)
 - **IXF** (*.ixf)
 - **XML** Extensible Markup Language file (*.xml)
 - **XLS** Microsoft Excel 97-2003 (*.xls)
 - **XLSX** Microsoft Excel 2007 (*.xlsx)
 - **PNG** (*.png)
6. Specify the location to save the object.
Note: The file will be exported if the **Root output directory** is specified by administrator.
There are two options:
 - a) In case the **Root output directory** is specified on the **Server-Side File System** page of the **Preferences** window, you can just type a path and a file name or only a file name in the **Path** field. The file will be exported to this root directory automatically. In QMF for WebSphere you can specify only the path, lying in the root directory set by administrator. Otherwise, the export will be forbidden and an error will occur.
 - b) To specify a conditional save location for the object, click **Formula** and then click the ellipsis (...). In the **Formula editor**, specify the formula to use to determine the location to save the object and click **OK**.
 7. Select the **Delete after task completion** check box if you want to remove the exported file from the specified location when the scheduled task is completed.
 8. Select the **Mark as attachment** check box if you want to be able to attach this query result set to the email automatically.
 9. Click **Next**.
 - If you selected the Text or CSV export formats, the **Specify [file type] export options** page opens. On this page you can select additional options that are available depending on the type of file

that you are creating. Then, the **Specify delimiter options** page opens. On this page, specify the detailed information about how the result set will be displayed in the selected format.

- If you selected any other export format except XML, the **Specify [file type] export options** page opens. On this page you can select additional options that are available depending on the type of file that you are creating.
- If you selected the XML format, click **Finish** instead of clicking **Next**.

10. Click **Finish**.

The **Operation Wizard** closes.

Related tasks

[“Setting the directory location preference for objects generated by Export actions” on page 17](#)

You can set preferences in QMF so that objects generated from various QMF actions are exported to a specific location on either the local file system or the server file system.

Related reference

[“Specifying file export options” on page 201](#)

You must specify one or more of the following options when you export query results to a file. The options that you specify depend on the type of file that you are exporting.

Specifying export query to mail operation options

You use the export query to mail operation to send query results in an email.

About this task

To add an export query to mail operation to an action:

Procedure

1. Open the **Specify operations** page of the **Action Wizard** and click **Add Operation**.
The **Operation Wizard** opens.
2. Click **Export query to mail** and then click **Next**.
The **Specify export parameters** page opens.
3. If the query has multiple result sets, select the result set that you want to export in the **Result set** field.
4. If the query has multiple display modes, select the display mode that you want to export from the **Display mode** list.
5. Select the file type that you want to export from the **Export type** drop-down list.
Valid file types are:
 - **Text** (*.txt)
 - **CSV** (*.csv)
 - **PDF** (*.pdf)
 - **HTML** (*.html)
 - **IXF** (*.ixf)
 - **XML** Extensible Markup Language file (*.xml)
 - **XLS** Microsoft Excel 97-2003 (*.xls)
 - **XLSX** Microsoft Excel 2007 (*.xlsx)
 - **PNG** (*.png)
6. Specify the name of the file attachment in the email in the **Attachment name** field.
7. If you want to specify a conditional file attachment name, click ellipses (...) next to the field.
The **Formula Editor** opens.
8. Specify the formula to use to determine the name of the attachment and click **OK**.

The **Formula editor** closes.

9. Click **Next**.

- If you selected any export format except XML, the **Specify [file type] export options** page opens. On this page, select additional options that are available depending on the type of file that you are creating and click **Next**.
- If you selected the XML format, the **Specify mail settings** page opens. Proceed from step “10” on [page 228](#).
- If you selected the Text or CSV export formats, the **Specify [file type] export options** page opens. On this page you can select additional options that are available depending on the type of file that you are creating. Then, the **Specify delimiter options** page opens. On this page, specify the detailed information about how the result set will be displayed in the selected format.

10. On the **Specify mail settings** page, specify the following information:

- From the **Format** list, select the format of the body text. The available formats are Text and HTML.
- Specify the recipient’s email address in the **To** field.
- Specify additional recipients in the **Cc** field.
- Specify the subject of the email in the **Subject** field.
- Specify the content of the email in the **Message** field.

Note:

If you want to include an embedded image in the text of the email, you must insert the following HTML snippet into the text:

```
</img>
```

where *cid*: is the standard abbreviated content identifier required when referencing an embedded image within a formatted email, and where *[image name]* is the name of the image file. For example, to embed an image file named `image1.png`, you would insert the following HTML snippet into the text of the email:

```
</img>
```

- Specify the email address from which the email is sent in the **From** field.

For each of the fields on the **Specify mail settings** page you can click the ellipses button (...) to access the Formula Editor. Use the Formula Editor to specify conditional values.

Note: The values that you set on this page can be specified dynamically during iterator work with the help of substitution variables. You can use an action group with iterator to send separate email messages to several recipients. To do so, you need to specify the iterator query that returns all necessary email information, define iterator variables, and type the variable names in the corresponding fields on the **Specify mail settings** page using the equality sign (=FORM, =SMTP). When you run the task, iterator goes over the iterator query records, extracts a set of email parameters at each iteration, and passes it to the *Export query to mail* operation.

11. To specify SMTP server parameters, click **SMTP settings**. In the SMTP Settings window, specify whether to use a server predefined by administrator or a custom one.

- a) If you decide to use a predefined SMTP server, select the required server from the list of predefined servers or click **Add Server** to create one.
- b) If you decide to use a custom server, specify server parameters including its address and port number. If the SMTP server requires authentication, select **Use SMTP authentication**, and then specify user name and password. If it is necessary to specify additional access preferences, click **Advanced** to open the **Advanced SMTP Settings** window where you can add required keywords supported by `javamail`. Click **OK**.

Click **OK** to close the **SMTP Settings** window.

12. Click **Finish** to close the **Operation** wizard.

Related reference

[“Specifying file export options” on page 201](#)

You must specify one or more of the following options when you export query results to a file. The options that you specify depend on the type of file that you are exporting.

Specifying print query operation options

You use the print query operation to print the specified query results.

About this task

To add a print query operation to an action:

Procedure

1. Open the **Specify operations** page of the **Action Wizard** and click **Add Operation**.
The **Operation Wizard** opens.
2. Click **Print query** and then click **Next**.
The **Specify printer** page opens.
3. Select the printer that you want to use to print the query results from the **Printer** drop-down list.
4. Click **Properties**.
The **Page Setup** window opens.
5. Select the size of the printable area from the **Size** drop-down list.
6. Select the paper tray to use from the **Source** drop-down list.
7. Select the orientation of the printout from the **Orientation** radio group.
8. Specify the size of the margins of the page in the **Margins** fields.
9. Click **OK**.
The **Page Setup** window closes and control returns to the **Operation Wizard**.
10. Click **Finish**.
The **Operation Wizard** closes.

Specifying export quick report to mail operation options

Use the export quick report to mail operation to send a quick report in an email.

About this task

To specify export quick report to mail operation options, perform the following procedure:

Procedure

1. Open the **Specify operations** page of the **Action Wizard** and click **Add Operation**.
The **Operation Wizard** opens.
2. Click **Export quick report to mail** and then click **Next**.
The **Specify export parameters** page opens.
3. Select the file type that you want to export from the **Export type** drop-down list.
Valid file types are:
 - **PDF** (*.pdf)
 - **HTML** (*.html)
4. Specify the name of the file attachment in the email in the **Attachment name** field.
5. If you want to specify a conditional file attachment name, click the ellipsis (...).
The **Formula Editor** opens.

6. Specify the formula to use to determine the name of the attachment and click **OK**.

The **Formula Editor** closes.

7. Click **Next**.

- If you selected the PDF format, the **Specify PDF export options** page opens. On this page, select additional export options that are available for the PDF format and click **Next**.
- If you selected the HTML format, the **Specify mail settings** page opens.

8. On **Specify mail settings** page, specify values for the following parameters:

- Select the format of the body text from the **Format** list. The available formats are Text and HTML
- The address that the email will be sent to in the **To** field
- The address that a carbon copy of the email will be sent to in the **Cc** field
- The subject of the email in the **Subject** field
- The body text of the email in the **Message** field
- The address that will send the email in the **From** field

For each of the fields on the **Specify mail settings** page you can click the ellipses button (...) to access the Formula Editor. Use the Formula Editor to specify conditional values.

9. To specify SMTP server parameters, click **SMTP Settings**. In the SMTP Settings window, specify whether to use a server predefined by administrator or a custom one.

- a) If you decide to use a predefined SMTP server, select the required server from the list of predefined servers or click **Add Server** to create one.
- b) If you decide to use a custom server, specify server parameters including its address and port number. If the SMTP server requires authentication, select **Use SMTP authentication**, and then specify user name and password. If it is necessary to specify additional access preferences, click **Advanced** to open the **Advanced SMTP Settings** window where you can add required keywords supported by javamail. Click **OK**.

Click **OK** to close the **SMTP Settings** window.

10. Click **Finish**.

The **Operation Wizard** closes.

Related reference

[“Specifying file export options” on page 201](#)

You must specify one or more of the following options when you export query results to a file. The options that you specify depend on the type of file that you are exporting.

Specifying export quick report operation options

Use the export quick report operation to export quick report results to a file.

About this task

To specify export quick report operation options, perform the following procedure:

Procedure

1. Open the **Specify operations** page of the **Action Wizard** and click **Add Operation**.
The **Operation Wizard** opens.
2. Click **Export quick report** and then click **Next**.
The **Specify export parameters** page opens.
3. Select the file type that you want to export from the **Export type** drop-down list.

Valid file types are:

- **PDF** (*.pdf)
- **HTML** (*.html)

4. Specify the location to save the object.

Note: The file will be exported if the **Root output directory** is specified by administrator.

There are two options:

- In case the **Root output directory** is specified on the **Server-Side File System** page of the **Preferences** window, you can just type a path and a file name or only file name in **Path** field. The file will be exported to this root directory automatically. In QMF for WebSphere you can specify only the path, lying in the root directory set by administrator. Otherwise, the export will be forbidden and an error will occur.
 - To specify a conditional save location for the object, click **Formula** and then click the ellipsis (...). In the **Formula editor**, specify the formula to use to determine the location to save the object and click **OK**.
5. Select the **Delete after task completion** check box if you want to remove the exported file from the specified location when the scheduled task is completed.
6. Select the **Mark as attachment** check box if you want to be able to attach this quick report to the email automatically.
7. If you selected the PDF format, click **Next**.

The **Specify PDF export options** page opens. On this page you can select additional options that are available depending on the type of file that you are creating.

8. Click **Finish**.

The **Operation Wizard** closes.

Related tasks

[“Setting the directory location preference for objects generated by Export actions” on page 17](#)

You can set preferences in QMF so that objects generated from various QMF actions are exported to a specific location on either the local file system or the server file system.

Related reference

[“Specifying file export options” on page 201](#)

You must specify one or more of the following options when you export query results to a file. The options that you specify depend on the type of file that you are exporting.

Specifying export visual report to mail operation options

You use the export visual report to mail operation to send a visual report in an email.

About this task

To specify export visual report to mail operation options, perform the following procedure:

Procedure

1. Open the **Specify operations** page of the **Action Wizard** and click **Add Operation**.
The **Operation Wizard** opens.
2. Click **Export visual report to mail** and then click **Next**.
The **Specify export parameters** page opens.
3. Select the file type that you want to export from the **Export type** drop-down list.
Valid file types are:
 - **PDF** (*.pdf)
 - **HTML** (*.html)
4. Specify the name of the file attachment in the email in the **Attachment name** field.
5. If you want to specify a conditional file attachment name, click the ellipsis (...).
The **Formula Editor** opens.
6. Specify the formula to use to determine the name of the attachment and click **OK**.

The **Formula Editor** closes.

7. Click **Next**.

The **Specify [file type] export options** page opens. On this page you can select additional options that are available depending on the type of file that you are creating.

8. On **Specify mail settings** page, specify values for the following parameters:

- Select the format of the body text from the **Format** list. The available formats are Text and HTML
- The address that the email will be sent to in the **To** field
- The address that a carbon copy of the email will be sent to in the **Cc** field
- The subject of the email in the **Subject** field
- The body text of the email in the **Message** field
- The address that will send the email in the **From** field

For each of the fields on the **Specify mail settings** page you can click the ellipses button (...) to access the Formula Editor. Use the Formula Editor to specify conditional values.

9. To specify SMTP server parameters, click **SMTP Settings**. In the SMTP Settings window, specify whether to use a server predefined by administrator or a custom one.

- a) If you decide to use a predefined SMTP server, select the required server from the list of predefined servers or click **Add Server** to create one.
- b) If you decide to use a custom server, specify server parameters including its address and port number. If the SMTP server requires authentication, select **Use SMTP authentication**, and then specify user name and password. If it is necessary to specify additional access preferences, click **Advanced** to open the **Advanced SMTP Settings** window where you can add required keywords supported by `javamail`. Click **OK**.

Click **OK** to close the **SMTP Settings** window.

10. Click **Finish**.

The **Operation Wizard** closes.

Related reference

[“Specifying file export options” on page 201](#)

You must specify one or more of the following options when you export query results to a file. The options that you specify depend on the type of file that you are exporting.

Specifying export visual report operation options

Use the export visual report operation to export a visual report to a file.

About this task

To specify export visual report operation options, perform the following procedure:

Procedure

1. Open the **Specify operations** page of the **Action Wizard** and click **Add Operation**.

The **Operation Wizard** opens.

2. Click **Export visual report** and then click **Next**.

The **Specify export parameters** page opens.

3. Select the file type that you want to export from the **Export type** drop-down list.

Valid file types are:

- **PDF** (*.pdf)
- **HTML** (*.html)

4. Specify the location to save the object.

There are two options:

- To specify a static save location for the object, click **Path** and enter the directory location where you want to save the object.

In case the **Root output directory** is specified on the **Server-Side File System** page of the **Preferences** window, you can just type a path and a file name or only a file name in the **Path** field. The file will be exported to this root directory automatically. In QMF for WebSphere you can specify only the path, lying in the root directory set by administrator. Otherwise, the export will be forbidden and an error will occur.

- To specify a conditional save location for the object, click **Formula** and then click the ellipsis (...). In the **Formula editor**, specify the formula to use to determine the location to save the object and click **OK**.
5. Select the **Delete after task completion** check box if you want to remove the exported file from the specified location when the scheduled task is completed.
 6. Select the **Mark as attachment** check box if you want to be able to attach this visual report to the email automatically.
 7. Click **Next**.
The **Specify [file type] export options** page opens. On this page you can select additional options that are available depending on the type of file that you are creating.
 8. Click **Finish**.
The **Operation Wizard** closes.

Related tasks

[“Setting the directory location preference for objects generated by Export actions” on page 17](#)

You can set preferences in QMF so that objects generated from various QMF actions are exported to a specific location on either the local file system or the server file system.

Related reference

[“Specifying file export options” on page 201](#)

You must specify one or more of the following options when you export query results to a file. The options that you specify depend on the type of file that you are exporting.

Adding a resource manipulation task

The resource manipulation action allows you to automatically copy, paste, delete, and email resources that are generated by the scheduled task.

About this task

To add a resource manipulation action to a scheduled task:

Procedure

1. Open a task from the **Task List** window.
2. Click **Add Action**.
The **Action Wizard** opens.
3. Specify a name for the action in the **Action name** field, click **Resource manipulation**, and then click **Next**.
The **Specify resource manipulation type** page opens.
4. Click **Copy** or **Send mail** and then click **Next**.
The next page of the wizard opens.
5. Specify the **Copy** or **Send email** options and click **Finish**.
The **Action Wizard** closes and control returns to the **Task** window.
6. Click **OK**.
The **Task** window closes and control returns to the **Task List** window.
7. Click **OK**.

The **Task List** window closes.

Related tasks

Adding an action group

You can use action groups to create a series of actions that should be performed successively.

Specifying variables for actions and action groups

You can create a scheduled task that runs a variety of data objects such as queries, quick reports, visual reports, and procedures containing parameters.

Scheduling local tasks

Use the specific to your operating system scheduler window or application schedule task window to schedule a task to run.

Sending HTML tables in email

You can send query results as HTML tables in the email when you work with scheduled tasks.

Related reference

Adding an execute object action

The execute object action allows you to automatically execute queries, a visual report, procedure, or quick report, and export, mail, or print it.

Specifying copy resource parameters

The copy resource manipulation action allows you to automatically copy, paste, and delete resources that are generated by the scheduled task.

About this task

To specify copy resource parameters:

Procedure

1. Open the **Specify resource manipulation type** page of the **Action Wizard**.
2. Click **Copy** and then click **Next**.

The **Specify copy resource parameters** page opens.

3. Specify the resource that you want to copy.

There are two options:

- To specify the static source location of the object, click **Path** and then click the ellipsis (...) at the bottom right of the **Source** area. In the **Open** window, to specify the static source location of the object, click **Path** navigate to the object you want to copy, select it and click **Finish**.
- To specify a conditional source location for the object click **Formula** and then click ellipsis (...). In the **Formula editor**, specify the formula to use to determine the location of the object and click **OK**.

4. Specify where you want to save the resource.

There are two options:

- To specify the static source destination of the resource, click **Path** and then click the ellipsis (...) at the bottom right of the **Source** area. In the **Save As** window, navigate to the location where you want to save the resource and its file name and click **Finish**.

In case the **Root output directory** is specified on the **Server-Side File System** page of the **Preferences** window, you can just type a path and a file name or only a file name in the **Path** field. The file will be exported to this root directory automatically. In QMF for WebSphere you can specify only the path, lying in the root directory set by administrator. Otherwise, the export will be forbidden and an error will occur.

- To specify a conditional source destination for the resource, click **Formula** and then click the ellipsis (...). In the **Formula editor**, specify the formula to use to determine where to save the resource and click **OK**.

5. To cut and paste the resource from the source location to the destination location, select the **Move** check box.
6. To create a new folder to contain the moved resource, select the **Create folders** check box.

Note: The **Create folders** check box is selected and cannot be cleared in QMF Server and QMF for WebSphere.

7. To delete the moved copy of the resource once the task has been completed, select the **Delete after task completion** check box.
8. Select the **Mark as attachment** check box if you want to be able to attach this resource copy to the email automatically.
9. Click **Finish**.

The **Action Wizard** closes and control returns to the **Task** window.

Related tasks

[“Setting the directory location preference for objects generated by Export actions” on page 17](#)

You can set preferences in QMF so that objects generated from various QMF actions are exported to a specific location on either the local file system or the server file system.

Specifying mail settings

The send mail resource manipulation action allows you to automatically send generated resources in an email.

About this task

To specify mail settings:

Procedure

1. Open the **Specify resource manipulation type** page of the **Action Wizard**.
2. Click **Send mail** and then click **Next**.
The **Specify mail settings** page opens.
3. Specify the format of the body text in the **Format** list. The available formats are Text and HTML.
4. Specify the recipient of the email in the **To** field.
5. Specify any additional recipients in the **Cc** field.
6. Specify the subject of the email in the **Subject** field.
7. Specify the text of the email in the **Message** field.

You can attach images as well as files of other available types.

If you want to include an embedded image in the text of the email, you must insert the following HTML snippet into the text:

```
</img>
```

where *cid*: is the standard abbreviated content identifier required when referencing an embedded image within a formatted email, and where *[image name]* is the name of the image file. For example, to embed an image file named `image1.png`, you would insert the following HTML snippet into the text of the email:

```
</img>
```

8. Specify the email address that will send the email in the **From** field.

For each of the fields on the **Specify mail settings** page you can click the ellipses button (...) to access the Formula Editor. Use the Formula Editor to specify conditional values.

9. To specify SMTP server parameters, click **SMTP Settings**. In the SMTP Settings window, specify whether to use a server predefined by administrator or a custom one.

- a) If you decide to use a predefined SMTP server, select the required server from the list of predefined servers or click **Add Server** to create one.
- b) If you decide to use a custom server, specify server parameters including its address and port number. If the SMTP server requires authentication, select **Use SMTP authentication**, and then specify user name and password. If it is necessary to specify additional access preferences, click **Advanced** to open the **Advanced SMTP Settings** window where you can add required keywords supported by javamail. Click **OK**.

Click **OK** to close the **SMTP Settings** window.

10. Click **Add**.

The **Enter a path as a text or formula** window opens.

11. Specify the path to the resource that you want the task to send in an email and click **OK**.

The **Enter a path as a text or formula** window closes.

12. In the **Attachments** area, select the required options from two lists.

- a) From the first list select the attachment that you want to add automatically. The available options are:
 - **No automatic attachments** not to attach any resource automatically.
 - **Attach marked only** to attach only the resources that were marked in the export or copy operations.
 - **Attach all** to attach all resources created in the export or copy operations.
- b) From the second list select the procedure that you want to apply to the attachments after the action execution. The available options are:
 - **Clear all automatic attachments** to delete all attachments sent automatically after the action execution.
 - **Clear automatic attachments in the group** to delete all automatic attachments, which are created for a group of actions with the sending an email action, after the action execution.
 - **Do not clear automatic attachments** to store all automatic attachments after the action execution.

13. Click **Finish**.

The **Action Wizard** closes and control returns to the **Task** window.

Adding an action group

You can use action groups to create a series of actions that should be performed successively.

Before you begin

Before creating an action group, you must first create a scheduled task.

About this task

To add an action group to a scheduled task:

Procedure

1. Open the **Task** window in one of the following ways:
 - Double-click a task in the **Tasks** list;
 - Select a task from the **Tasks** list, and then click **Edit Task**.
2. In the **Task** window, click **Add Action** to open the **Action** wizard.
3. Type a name in the **Action name** field.
4. Select **Action Group**, and then click **Finish**.

Related tasks

[Adding a resource manipulation task](#)

The resource manipulation action allows you to automatically copy, paste, delete, and email resources that are generated by the scheduled task.

[Specifying variables for actions and action groups](#)

You can create a scheduled task that runs a variety of data objects such as queries, quick reports, visual reports, and procedures containing parameters.

[Scheduling local tasks](#)

Use the specific to your operating system scheduler window or application schedule task window to schedule a task to run.

[Sending HTML tables in email](#)

You can send query results as HTML tables in the email when you work with scheduled tasks.

Related reference

[Adding an execute object action](#)

The execute object action allows you to automatically execute queries, a visual report, procedure, or quick report, and export, mail, or print it.

Adding actions to an action group

You can use drag-and-drop method to add actions to an action group.

Before you begin

You must create an action group.

You must create an action.

About this task

To add an action to an action group:

Procedure

1. Open the **Task** window in one of the following ways:
 - Double-click a task in the **Tasks** list;
 - Select a task from the **Tasks** list, and then click **Edit Task**.
2. In the Task window, select an action that you want to add to a group.
3. Drag the selected action onto an action group item.

What to do next

Next: [Setting iterator parameters](#)

Related tasks

[Adding an execute object action](#)

[“Adding a resource manipulation task” on page 233](#)

The resource manipulation action allows you to automatically copy, paste, delete, and email resources that are generated by the scheduled task.

Removing actions from an action group

You can use drag-and-drop method to remove actions from an action group.

Before you begin

You must create an action group and populated it with actions.

About this task

To remove an action from an action group:

Procedure

1. Open the **Task** window in one of the following ways:
 - Double-click a task in the **Tasks** list;
 - Select a task from the **Tasks** list, and then click **Edit Task**.
2. In the Task window, expand an action group that you want to edit.
3. Select an action that you want to remove from the action group.
4. Drag the selected action outside the action group.

Setting iterator parameters

You can use iterator to run an action group for several times in a cycle. You use the **Iterator Parameters** window to manage iterator preferences.

Before you begin

You must create an action group.

About this task

To set iterator parameters:

Procedure

1. In the **Actions** list of the **Task** window, right-click the action group that you want to run in a cycle, and then select **Set Iterator** from the pop-up menu.
2. In the **Iterator Parameters** window, click the ellipsis (...) to specify a query for iterator.

The columns of this query define iterator variables and the number of rows specifies the number of iterations.
3. In the **Open** window navigate to and select an object that you want to use as a query for iterator.

The following objects can be used as a query:

 - Charts
 - Visual Queries
 - Analytical Queries
 - Dynamarts
 - Compound Dynamarts
 - Tables
4. Click **Finish**. The path to the selected object is displayed in the **Query path** field.
5. The data source that contains the selected object is added to the **Data Sources** list. You can specify your login information for a data source in the **Login** and **Password** fields.
6. To add another data source to the list, perform the following procedure:
 - a) Click **Add Data Source** to open the **Add Data Source** window.
 - b) From the **Data Source** list, select a data source that you want to add.
 - c) Type your login information in the **User name** and **Password** fields.
 - d) Select the **Remember the password** check box if you want QMF to remember the password for the user ID that you entered in the **User name** field.
7. To remove a previously added data source from the list, select this data source, and then click **Remove Data Source**.

8. In the **Variables** list, link the query columns with the prompts used in the source object that you specify in Execute object actions. Click in the **Prompt name** cell and select the prompt from the drop-down list.
9. Use the **Continue on failure** check box to specify whether to continue iterator work when an error occurs. If the check box is selected and an error occurs within any iteration, iterator skips this iteration and continues working. If an error occurs and the check box is cleared, iterator stops working.

Note: All error messages can be viewed in the **Output** view.

10. Click **OK** to close the **Iterator Parameters** window.

What to do next

Next: [Managing prompts in an action group](#)

Related tasks

[Adding an execute object action](#)

Managing prompts in an action group

You can create a scheduled task that runs a variety of data objects such as queries, quick reports, visual reports, and procedures containing parameters.

About this task

You use the **Manage Shared Prompts** window to define how the parameter values must be passed to the source objects specified in the task.

Note: Depending on the type of an object, parameters are represented by different elements. For queries, parameters are prompts defined for these queries. For quick reports and visual reports, parameters are prompts specified in queries on which these reports are based. For procedures, parameters are variables defined in these procedures.

You can group similar prompts used in the source objects so that all the prompts within one prompt group receive the value passed to this group. By default, there are as many groups as prompts in the source objects within the action group. Each group stores one particular prompt and borrows its name. You can move prompts between the groups. If you move the last prompt from the group, this group is removed automatically because empty groups are not allowed.

You must create an action group and populated it with execute object actions.

To manage shared prompts:

Procedure

1. Open the **Task** window by selecting a task from the **Tasks** list and clicking **Edit Task**.
2. In the **Task** window, right-click an action group, and then select **Manage Prompts**.
3. From the **Available prompts** list in the **Manage Shared Prompts** window, select a prompt that you want to add to a group.
4. From the **Shared prompts** list, select a group that you want to add the prompt to.
5. Click **Add to Group**. The selected prompt is added to the prompt group.
6. To remove a prompt from a prompt group, select a prompt from the **Shared prompts** list, and then click **Detach Prompt**.
7. To rename a prompt group, select the prompt group that you want to rename, and then click **Rename prompt group**.
8. In the **Prompt Group** window, specify the name and string to display, and then click **OK**.
9. To change the order of the prompt groups in the **Shared prompts** list, select the node of the prompt group or a prompt in the group, and click **Move Prompt Group Up** or **Move Prompt Group Down**.

10. Add an iterator query for the action group and link the query columns with the parameters used in the source objects that you specified in actions within the action group.

Related tasks

[“Working with prompts in queries” on page 76](#)

You can filter your queries or dynamarts by adding prompts that direct the query or dynamart to focus on specific values or ranges. This limits the amount of data to be queried and provides more focused query results. The filtering works on the database level.

Specifying variables for actions and action groups

You can create a scheduled task that runs a variety of data objects such as queries, quick reports, visual reports, and procedures containing parameters.

About this task

You use the **Set Variables** window to define how the parameter values must be passed to the source objects which are specified in the task. You can also use the specified parameters in actions themselves as substitution variables. For example, you can form the file system path by using parameters.

Note: Depending on the type of an object, parameters are represented by different elements. For queries, parameters are prompts defined for these queries. For quick reports and visual reports, parameters are prompts specified in queries on which these reports are based. For procedures, parameters are variables defined in these procedures.

You can set parameter values for both action groups and single actions. You can also set the iterator for the same group. The parameters provided by the single action take precedence, then goes parameter values set for the group, then parameter values set by the iterator.

To specify parameter values:

Procedure

1. Open the **Task** window by selecting a task from the **Tasks** list and clicking **Edit Task**.
2. In the **Task** window, right-click the action or action group that you want to work with and select **Set Variables** from the menu. All the parameters specified for the selected data object are displayed in the list of parameters in the **Set Variables** window.

Note: If you decide to set parameter values to an action group to which you have already assigned shared prompts, the specified shared prompts are displayed instead of parameters defined in the source object.

3. To add a parameter, click **Add Variable**, and then specify its name and the value in the **Add Variable** window.
4. To edit or remove a variable, select the variable you want to edit or remove, and click **Edit Variable** or **Remove**.
5. To set the parameter value, open the **Add Variable** or **Edit Variable** window and enter either the constant value or a formula in the **Value** field. To open the **Formula Editor** window, click the ellipsis (...) next to the **Value** field.

Note: You can also set the parameter value directly in the **Set Variables** window by simply typing it in the required cell of parameters table.

6. You can change the order in which parameters are calculated by clicking **Move Up** and **Move Down**. Parameters are calculated sequentially as they appear in the list and those that have already been calculated can be used to calculate other parameters.
7. Click **OK** to close the **Set Variables** window.

Related tasks

[Adding a resource manipulation task](#)

The resource manipulation action allows you to automatically copy, paste, delete, and email resources that are generated by the scheduled task.

Adding an action group

You can use action groups to create a series of actions that should be performed successively.

Scheduling local tasks

Use the specific to your operating system scheduler window or application schedule task window to schedule a task to run.

Sending HTML tables in email

You can send query results as HTML tables in the email when you work with scheduled tasks.

Related reference

Adding an execute object action

The execute object action allows you to automatically execute queries, a visual report, procedure, or quick report, and export, mail, or print it.

Scheduling local tasks

Use the specific to your operating system scheduler window or application schedule task window to schedule a task to run.

Before you begin

You must have a task to schedule.

About this task

If you select **Native** on the **General** tab in the **Preferences** window, QMF integrates with the built-in scheduler of the operating system to schedule tasks. For more information about the built-in scheduler that is available, check your operating system documentation, or use the context help in the scheduler window.

Restriction: If **Native** is selected, scheduled tasks are not run on schedule in the 64-bit application installed in Windows 7 or Windows Vista when User Account Control is disabled.

If you select **QMF for Workstation scheduler** on the **General** tab in the **Preferences** window, the application schedule task window opens.

To schedule a task with QMF for Workstation scheduler:

Procedure

1. Select **View > Scheduled Tasks**.

The **Task List** window opens.

2. On the **Local tasks** tab, select the task that you want to schedule and click **Schedule Task**.

Restriction: When connected to a Derby database, local tasks have the following restrictions:

- You cannot execute several local tasks at the same time.
 - Local tasks are not executed when the application is running.
3. Specify the operating system user name and password in the **User** and **Password** fields.
 4. Optional: In the **Working directory** field, specify the directory for temporary files, which are created during the task execution.
 - Select the **Delete when done** check box to delete the task from the scheduler window after its execution.
 - Select the **Run only if logged on** check box to run the task only if the user with the credentials that were specified in the **User** and **Password** fields is logged on the operating system.
 - Select the **Disabled** check box to disable the task execution.

5. Specify the schedules that you want to apply to the task in the **Schedules** list.
6. To add a schedule, click **Add Schedule**.
7. Optional: To remove a schedule, which you do not need, click **Remove Schedule**.
8. Select the execution frequency of the task from the **Type of schedule** list.
9. Specify the frequency details in the area next to the **Type of schedule** list. The options vary depending on the selected type of schedule.
10. Click **OK** to close the scheduler window.

Related tasks

Adding a resource manipulation task

The resource manipulation action allows you to automatically copy, paste, delete, and email resources that are generated by the scheduled task.

Adding an action group

You can use action groups to create a series of actions that should be performed successively.

Specifying variables for actions and action groups

You can create a scheduled task that runs a variety of data objects such as queries, quick reports, visual reports, and procedures containing parameters.

Sending HTML tables in email

You can send query results as HTML tables in the email when you work with scheduled tasks.

Related reference

Adding an execute object action

The execute object action allows you to automatically execute queries, a visual report, procedure, or quick report, and export, mail, or print it.

Sending HTML tables in email

You can send query results as HTML tables in the email when you work with scheduled tasks.

Before you begin

Make sure that the selected format of a letter is HTML and the table you want to add is saved in the HTML format too.

About this task

The inserted tables appear in the body of the letter in HTML format.

Adding an HTML table is available only for the following tasks of the scheduler:

- Exporting query to mail task. As a result of this task you get query result set in HTML format, which can be sent in the email.
- Sending mail resource manipulation task. In this task you can insert the necessary HTML tables created when performing other tasks.

To insert an HTML table in the body of an HTML letter:

Procedure

1. On the **Specify mail settings** page of the **Operation Wizard** window, click an ellipsis button (...) in the **Message** field. The **Formula Editor** opens.
2. From the list of the functions expand the **Scheduler** and select **insertHTMLTable** or type the name of this function in the **Filter** field.
3. Double-click the function and specify its parameters.
4. Click **OK** to add the function and close the **Formula Editor**. The specified function is added to the **Message** field.

Example

You can add *grid.html* table to the letter with the following parameters:

- When you work with export task, just add the function without any parameters

```
=insertHTMLTable()
```

- When you work with resource manipulation task, you can specify only one parameter, in case the exported HTML file has UTF-8 encoding.

```
=insertHTMLTable("grid.html")
```

- You must specify both parameters if the encoding of the exported HTML file differs from UTF-8.

```
=insertHTMLTable("grid.html", "UTF-16")
```

Related tasks

[Adding a resource manipulation task](#)

The resource manipulation action allows you to automatically copy, paste, delete, and email resources that are generated by the scheduled task.

[Adding an action group](#)

You can use action groups to create a series of actions that should be performed successively.

[Specifying variables for actions and action groups](#)

You can create a scheduled task that runs a variety of data objects such as queries, quick reports, visual reports, and procedures containing parameters.

[Scheduling local tasks](#)

Use the specific to your operating system scheduler window or application schedule task window to schedule a task to run.

Related reference

[Adding an execute object action](#)

The execute object action allows you to automatically execute queries, a visual report, procedure, or quick report, and export, mail, or print it.

Chapter 11. Generating classic and quick reports

Report developers can select from one of several formats that can be used to generate a report for a set of query results.

The available reporting formats are:

- Text-based, tabular classic reports that are backward-compatible with the legacy QMF infrastructure.
- Graphical visual reports authored using a built-in visual designer.

Note: You can navigate a classic report using the **arrow**, **Home**, and **End** keys on the keyboard. The **Up** and **Down** arrows scroll through the pages of the report vertically. The **Left** and **Right** arrows scroll through a page horizontally. The **Home** key navigates back to the top of the first page of the report. The **End** key navigates to the end of the last page of the report.

Creating a new report

You can create a new report using the **Display Report** wizard.

About this task

To generate a report in any of the formats:

Procedure

1. Select **Results > Display Report**. The **Display Report** wizard opens.
2. From the first page of the wizard, select whether you will be creating a new report or using an existing report. Your choices are:
 - Create a new report
 - Use an existing report stored in a file
 - Use an existing report stored in the QMF catalog
 - Use an existing report stored in the repository
3. Click **Next**.
4. Depending on your selection one of the following dialogs opens:
 - If you selected **Create a new report**, the second page of the wizard requests that you specify what type of report you will create. The available reporting options are:
 - **Create a classic report** to create a classic report.
 - **Create a Visual report** to create a visual report.

For each type of report you must specify whether the report will be generated based on how the current query is formatted or on simple default formatting. Select **Create from query** to specify that the report will be generated based on how the current query results are formatted. Select **Default** to specify that the report will be generated on simple default formatting. For classic report you must specify the **Fetching options**. Select **Use available data** to include in the report only the data that has already been fetched. Select **Fetch all data** to complete the process of fetching the data before the report is generated so that all data is included.

- If you selected **Use an existing report stored in a file**, the second page of the wizard requests that you specify the path to the file that contains the report in the **Name** field.
- If you selected **Use an existing report stored in the QMF catalog**, the second page of the wizard requests that you specify the database where the QMF catalog resides in the **Database** field; and the report owner, name and type in their respective fields. To see a list of reports click **List Objects**.

- If you selected **Use an existing report stored in a repository**, the second page of the wizard requests that you select from the repository or a repository workspace the report that you want to open.
5. Click **Finish**. The report is displayed in the active editor window.

Selecting a classic report's data source object

You can select a new data source object for the current classic report or specify that a form will always use a specific data source object to produce a classic report.

About this task

Changing data source objects by using the **Data Source Object** dialog makes it easy to view different sets of query results data in the same classic report format. QMF will run the newly selected data source object, format the query results using the current form, and display the updated report in the active editor window.

Specifying that a particular data source object will always be used when a form is opened streamlines the steps required to create a classic report. If specified, when QMF opens such form, it will find the assigned data source object, run the object, format the query results according to the specifications of the form and display the generated classic report in the editor window. If you do not specifically assign a data source object to a form, then when you open that form, QMF formats the currently active query results into a report. If you want specific query result data for the report, you must find the object, run the object and then generate the report.

To specify a data source object:

Procedure

1. Open the Data Source Object window.

With a form in the active window, select **Form > Data Source Object**.

2. Specify where the data source object is located by selecting one of the following options:
 - Click **From file** to specify that the data source object is saved to a file located on a local or network drive. The required fields become available. Specify the name of the file in the **Name** field. You can search the list of available files by clicking the **Browse** button.
 - Click **From data source** to specify a data source object that is stored on a database. The required fields become available. Specify the data source where the object is stored in the **Data Source** field. Specify the owner, name, and type for the data source object in the **Owner**, **Name**, and **Type** fields. You can click the **List Objects** button to search the database for an object.
 - Click **From repository** to specify a data source object that is saved in a repository. The required fields become available. Expand repository directories and folders to expose the data source object that you want to select.
 - Click **From open object** to get the data source object from one of the currently open query objects. Select the name of the object from the **Title** list box of the open query objects.
3. If the data source object that you have selected produces multiple result sets, as in the case with stored procedures or multiple SQL statement queries, specify the index number of the result set that you want to use in the **Result set** field.
4. Check the **Always use this object when opening this document** check box to specify that whenever the currently active form is opened, the object that you specified will be used as the data source object. This option is not available when the selected data source object is a currently open object.
5. Click **OK** to generate the report using the selected data source object. If you specified that the currently selected data source object always be used when this form is opened, the next time you open the form the object will be run, and the query results will be formatted according to the specifications of the form into a report.

Listing data source objects

You use the **Object list** window to list the objects that have been saved to the data source.

About this task

You can select an object from this list to be used as the data source for a report.

To list the objects on a data source:

Procedure

1. Open the **Data Source Object** window.
With a form in the active window, select **Form > Data Source Object**.
2. Select **From data source** radio button and then click **List Objects**.
3. From the **Data Source** drop-down list, select the name of the data source whose objects you would like to list.
4. In the **Owner** field, enter the owner of the objects that you want to list. You must specify the owner in the correct format for identifiers.
5. In the **Name** field, enter the name of the object that you want to list. You must specify the name in the correct format for identifiers. You can specify a matching pattern to match multiple names.
6. In the **Folder** field, specify the name of the folder that stores the object that you want to list. Click the ellipsis (...) to specify any other folder on this server.
7. Click **Refresh List** to initiate a search for the objects on the database that match the search criteria that you specified in the **Owner** and **Name** fields. Only those objects that you are authorized to use are included in the list. The results are returned in the **Object List** table.
8. From this list, select the object that you want to use as the classic report data source object.
9. Click **OK**. The **Object List** window closes. Control returns to the **Data Source Object** window.

Exporting reports to a file

You can export a classic report, quick report, or a visual report to a file. If your report is a classic one, you can export it as a text, HTML or PDF file. If your report is a quick one or a visual one, you can export it as an HTML or a PDF file.

About this task

To export a report:

Procedure

1. Open the **Export Report** wizard in one of the following ways:
 - With a report in the active window, select **File > Export** to open the **Select** page of the **Export** wizard, and then select **Report** from the list of available export wizards.
 - With a report in the active window, click the **Export** toolbar button.
2. Select the report that you want to export from the **Reports** list box.
3. If you are using QMF for Workstation, specify where the report file should be saved in the **Report file** field. You can use the **Browse** button to search for the location.
4. Select the type of file that you are exporting. You can export a classic report as a text file (*.txt), an HTML file (*.htm;*.html), or a PDF file (*.pdf). You can export a quick report and a visual report as an HTML file (*.htm;*.html) or a PDF file (*.pdf).
5. If you are exporting a PDF file, click the **Next** button. The **Set up PDF export options page** opens. Select the **Add Document Information** check box and specify character strings in the **Title**, **Author**, **Subject** and **Keywords** fields to add document information to the PDF. Select the **Use only ANSI fonts** check box to use only ANSI fonts in the PDF.

Note: Some of the fonts (for example, TTC) can be displayed incorrectly in PDF files.

6. If you want to export your report as an HTML file, select HTML from the **Export type** list and click **Next**. The **Set up file export options** page opens. Select the **Split report into separate pages** check box to save each page of your report individually.
7. Click **Finish**.
 - If you are using QMF for Workstation, the report is exported to a file in the format that you selected.
 - If you are using QMF for WebSphere, a **Save** window opens that lists the individual files for download, and an archived ZIP file (*.zip) that contains all of the files for download. Click one or more of the hyper-links to download the files. When you have finished downloading the files, click **OK**.

Creating classic report forms

Classic reports are generated using query results as the data source and a form template. The form template, referred to as a "form", specifies how you want the query results formatted in the report.

About this task

Forms are considered objects and they can be saved in your repository, in the QMF catalog, or in a file. When you open a form object that has been saved, you are actually running the form object to generate the report. When opened, forms automatically use the currently active query results as the data source.

Alternatively, you can specify that a specific set of query results data always be used with a specific form to create a report. If you select this option, when the form object is opened a specific query object is run to obtain the requested query results and then those query results are formatted into a report using the form specifications.

You create and edit forms using the **Design** page of the **Classic Report editor** window. Reports are displayed in the **Report** page of the **Classic Report editor** window.

Designing a classic report form

You can create and edit forms using the **Design** page of the **Classic Report editor** window. Generated reports are displayed in the **Report** page.

About this task

To create a form:

Procedure

1. Open the **Design** page for a classic report in one of the following ways:
 - Select **File > New > Form**. The **Create New Form** wizard opens. Specify the name of the form in the **Name** field and where the form will be saved in the **Data source** field. Click **Finish**.
 - Select the **New Form** toolbar button. Click the **Design** tab to open the **Design** page of the **Form** window.
2. The **Design** page is divided into two sections. To the left is the **Form structure** tree. Each node in the tree represents a section of the report. As you select a node from the tree, the fields that you have to specify for that section of the report are displayed.
3. To design your form, select one or more sections from the **Form structure** tree and enter the corresponding fields. A classic report includes the following sections:
 - Main
 - Breaks
 - Calculations
 - Columns
 - Conditions

- Details
 - Final
 - Options
 - Page
4. When you have completed designing the form, you can save the form in your repository, in the QMF catalog, or to a file.

Specifying Main formatting options

The formatting fields contained within the **Main** section of a report provide the formatting instructions for the most common elements of a report including columns, form headings, footings, and breaks.

About this task

Some of the elements that you define in the **Main** section of the report can also be defined in other sections.

| Element | Description |
|--------------------------------------|---|
| Total width of report columns | This property displays the total width of the tabular data in the report. |
| Column properties | In the table at the top of the page, you can specify which columns to display and set their properties. You can use the buttons above the table to add or remove columns, open Edit column dialog, and change the columns order. See Column properties for details. |
| Page | Use the Heading text and Footing text fields to specify the heading and footing of the page. |
| Final | Use the Text field to specify the final line text. |
| Breaks | Use the following elements to specify breaks properties: <ul style="list-style-type: none"> • New page for break Check this check box to start a new page in the report when the current break level occurs. • Footing text Use this field to specify the break footing text. |
| Outlining for break columns | Break columns are the columns with the Usage code set to Break. The report lines are grouped by the value in a break column. If you check this check box the value is displayed only in the first line of a group. Otherwise, the value is displayed in every line of a group. |
| Default break text (*) | Check this check box to set asterisk as the default break text. If there is no aggregation set for any of the columns, an asterisk is shown at the end of the break line. |

Column properties

You can edit the column properties in the table or by using the Edit column dialog. Use the following fields to specify column properties.

- **Num:**

Use this property to identify the column. This number is used to resolve form variables, and to determine how query result columns map to report columns.

Enumeration starts at 1. You can change the column number by moving it up or down in the columns list. For this, use the Move Up Column and the Move Down Column buttons.

- **Heading:**

Use this field to specify the column heading, up to 40 characters in length. Use the underscore character (`_`) to create a multiline heading. Leading underscores are ignored. For example, `FIRST_SECOND` is displayed as:

```
FIRST
SECOND
```

- **Usage or Usage code:**

Use this field to assign a usage code to the column. Usage codes provide summary information about the data in a column. For example, usage codes can provide total summary information at the end of a column, or partial summaries at control breaks in a table. The usage codes available depend on the data in the column and the type of summary. From the drop down list of available usage codes, select the appropriate one. Refer to the online help system for more information on each usage code.

- **Indent:**

Specify the number of blank spaces to the left of the column. The range is between 0 and 999.

- **Width:**

Use this field to specify the width of the column. The number can be between 1 and 32,767. Make sure you allow enough width for currency symbols, the negative sign, and other special characters. If the column is too narrow for numeric data, it is replaced with asterisks. If the column is too narrow for character or date/time data, it is truncated based on the alignment specified. You can use a wrapping edit code to have the data wrap within the column width.

- **Edit:**

Select an edit code to control how data is formatted in a report. Edit codes are available for character, date, graphic, numeric, time, timestamp data. Also, user-defined edit codes can be created.

From the drop down list of available edit codes, select the appropriate one. Refer to the online help for more information about edit codes.

- **Seq:**

Use this field to specify the sequence of the columns in the report. If two columns have the same sequence number, they appear in the report in the column number order.

Note: The **Seq** field is ignored in reports that use the ACROSS usage code or if the Automatic reordering of report columns is selected on the **Form Options** window.

Specifying Breaks formatting options

The formatting elements contained within the **Breaks** section of a report specify the characteristics, content, and placement of up to six subtotal lines in a report, along with break heading text and break footing text.

About this task

| <i>Table 24. Breaks section elements for a classic report</i> | |
|---|---|
| Element | Description |
| New page for break | Check this check box to start a new page in the report when the current break level occurs. |

Table 24. Breaks section elements for a classic report (continued)

| Element | Description |
|-----------------------------------|--|
| Repeat detail heading | Check this check box to repeat the current detail heading with the current break level. |
| Blank lines before heading | Use this field to specify the number of blank lines to be included before the first break heading line. This value must be a number between 0 and 999. |
| Blank lines after heading | Use this field to specify the number of blank lines that are to be included after the last break heading line. This value must be a number between 0 and 999. |
| Break heading text | <p>You use the buttons above the Break heading text table to add, edit, or remove lines, and also change the lines order.</p> <p>Use the following fields to specify the break heading text:</p> <ul style="list-style-type: none"> • Line: Use this field to identify lines of break heading text and their relative positions to each other in the report. You can specify any number from 1 to 999 or a blank. You do not have to start with line one and you do not have to specify consecutive line numbers. If you use the same line value for more than one line, those lines are joined according to the value specified in the Alignment field. Lines with the same value in the Line field overlay each other if they are longer than the report width, or if their values in the Alignment field conflict. • Alignment: Use this field to specify where horizontally on the line this text is placed. Valid alignment values are LEFT, RIGHT, CENTER, APPEND, or the number of a specific position in the line. If you specify APPEND, the text is placed immediately following the previous line of the text. • Text: Use this field to specify the text that appears in the break heading. You can use form variables and global variables in this field. |
| New page for footing | Check this check box to start a new page in the report when the current break level ends. |
| Put break summary at line | Use this field to specify where vertically in the break footing lines, the break summary line is placed. This value must be a number between 1 and 999. If you do not want a summary line, specify NONE. |
| Blank lines before footing | Use this field to specify the number of blank lines to be included before the first break footing line. This value must be a number between 0 and 999, or BOTTOM which is treated as the value 0 when producing reports. |
| Blank lines after footing | Use this field to specify the number of blank lines that are to be included after the last break footing line. This value must be a number between 0 and 999. |

Table 24. Breaks section elements for a classic report (continued)

| Element | Description |
|---------------------------|--|
| Break footing text | <p>You use the buttons above the Break footing text table to add, edit, or remove lines, and also change the lines order.</p> <p>Use the following fields to specify the break footing text:</p> <ul style="list-style-type: none"> • Line: Use this field to identify lines of break footing text and their relative positions to each other in the report. You can specify any number from 1 to 999 or a blank. You do not have to start with line one and you do not have to specify consecutive line numbers. If you use the same line value for more than one line, those lines are joined according to the value specified in the Alignment field. Lines with the same value in the Line field overlay each other if they are longer than the report width, or if their values in the Alignment field conflict. • Alignment: Use this field to specify where horizontally on the line this text is placed. Valid alignment values are LEFT, RIGHT, CENTER, APPEND, or the number of a specific position in the line. If you specify APPEND, the text is placed immediately following the previous line of the text. • Text : Use this field to specify the text that appears in the break footing. You can use form variables, HTML variables, and global variables in this field. |

Using HTML variable within the Text field

You can insert HTML variables in **Text** fields in order to format the text data for HTML presentation. You can place these HTML variables in any section of a form.

About this task

Table 25. HTML variables

| HTML Variable | Use this variable to... |
|---------------------|---|
| IMAGE _n | Places the contents of column 'n' inside an HTML image reference tag, making graphic files visible in reports. |
| LINK _n | Places the contents of column 'n' inside an HTML link tag, which is used to add links to other HTML pages. |
| MAILTO _n | Places the contents of column 'n' inside an HTML mail to tag for an email link. |
| ANCHOR _n | Places an anchor point at the location of the variable value. These anchors are referenced from within the report, such as the top of the page or the beginning of a break section. |
| REF _n | Places the text contained in the DSQQW_HTML_REFTXT (a global variable, which contains text that appears in a report; the default = 'BACK TO') into an HTML reference tag, and is used to create links to anchors created with the ANCHOR _n variable in the report. |

Specifying Calculations formatting options

You use the **Calculations** options to define report calculation expressions.

About this task

You can specify the Calculations formatting options in the **Calculations** table. Use the buttons on the interface to add, edit or remove Calculations, and also change the Calculations order.

Use the elements described in the following table to set Calculation properties.

| Element | Description |
|--------------------|---|
| ID | Specify a number, between 1 and 999, that identifies the calculation expression. |
| Expression | Type a valid expression, up to 50 characters. Form variables cannot be used in this field. |
| Pass nulls? | Specify whether to pass the expression for evaluation when a variable in the definition has a null value. |
| Width | Specify the width used to format the result of the calculated expression when used as a variable in the form. Click in the column and specify the needed value by using the spin button. |
| Edit | Specify the edit code, which is used to format the result of this calculated expression when used as a variable in the form. Click in the column to open a drop down list of available edit codes and select the needed one. See the online help for more information about edit codes. |

Specifying Columns formatting options

You use the **Columns** formatting options to control the appearance and formatting of columns in the report.

About this task

You can specify the Columns formatting options in the table in the **Columns** section of the report. You use the buttons above the table to add or remove columns, open Edit column dialog, and change the columns order.

You can edit the column characteristics in the table or by using the **Edit column** dialog.

Definable characteristics include column heading, column usage, indentation, width, edit code, and sequence. The **Total width of report columns** property displays the total width of the tabular data in the report.

Columns formatting options that you define in the **Columns** section of the report can also be defined in the **Main** section.

Table 27. Columns option elements for a classic report

| Element | Description |
|----------------|--|
| Num | Specifies the number that identifies the column. This property is used resolve variable references, and to determine how query result columns map to report columns. Enumeration starts from 1. You can change the column number by moving it up or down in the columns list. For this, use the Move Up Column and the Move Down Column buttons. |
| Heading | Use this field to specify the column heading, up to 40 characters in length. Use the underscore character to create a multiline heading. Leading underscores are ignored. For example, FIRST_SECOND is displayed as: <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc; width: fit-content;"> FIRST SECOND </div> |
| Usage | Specify a usage code that will be used to collect summary information about the data in a column, such as total summary information at the end of a column, or partial summaries at control breaks in a table. The usage codes that are available depend on the data in the column and the type of summary. See the online help for more information about usage codes. |
| Indent | Specifies the number of blank spaces to indent from the left of the column. The range is between 0 and 999. |
| Width | Specify the width of the column. The range is between 1 and 32767. If the column is too narrow for numeric data, the data is replaced with asterisks. If the column is too narrow for character or date/time data, the data is truncated (based on the alignment specified). To avoid this, you can use a wrapping edit code to have the data wrap within the column width. See the online help for more information about edit codes. |
| Edit | Specify an edit code, which controls how data is formatted in a report. Edit codes are available for character, date, graphic, numeric, time, timestamp data. Also, user-defined edit codes can be created. See the online help for more information on edit codes. |
| Seq | Specifies the sequence number for the columns in the report. If two columns have the same sequence number, they appear in the report in column number order, in the Num field. Note: The sequence number is ignored in reports using the ACROSS usage code or if you check the Automatic reordering of report columns check box. |

Specifying Conditions formatting options

You use the **Conditions** formatting options to define conditional formatting constraints, such as suppressing records that do not meet certain characteristics.

About this task

You can specify the Conditions formatting options in the **Conditions** table. You use the buttons on the interface to add, edit, or remove condition lines, and change the condition lines order.

Use the elements described in the following table to set condition properties.

| Element | Description |
|-------------------|---|
| ID | Specify a number, between 1 and 999, that identifies the conditional expression. |
| Expression | Type a valid expression, up to 50 characters. An expression that evaluates to 1 is considered true; anything else is considered false. Form variables cannot be used in this field. |
| Pass nulls | Specify whether to pass the expression for evaluation when a variable in the definition has a null value. |

Specifying Details formatting options

You use the **Details** formatting fields to define report detail headings and body text.

About this task

This is where you can combine or replace tabular data with free-form text to create form letters or address labels. Each detail variation of headings and text can correspond to a condition specified using the **Conditions** node.

When you choose the **Details** node in the Form structure tree, the **Add Detail** and **Remove Detail** buttons appear above the tree. By using these buttons you can add or remove Details.

| Element | Description |
|---|--|
| Enable | Specify when to enable the detail variation: <ul style="list-style-type: none">• YES indicates that the variation is always selected for formatting.• NO indicates that the variation is never selected for formatting. You can conditionally enable the variation by referring to a Form Conditions expression using the Cnn and Enn specifications. |
| Include column headings with detail headings | Check this check box to display column headings with detail headings. |

Table 29. Detail section elements for a classic report (continued)

| Element | Description |
|----------------------------------|--|
| Detail heading text | <p>You use the buttons above the Detail heading text table to add, edit, or remove lines, and also change the lines order.</p> <p>Specify the lines of detail heading text:</p> <ul style="list-style-type: none"> • Line: specify where the text appears vertically in the detail heading. You do not have to start with line 1 and you do not have to specify consecutive line numbers. • Alignment: specify where the text is placed horizontally on the line. Valid alignment values are LEFT, RIGHT, CENTER, APPEND, or the number of a specific position in the line. If you specify APPEND, the text is placed immediately following the previous line of text. • Text: type the text that appears in the detail heading. You can use form variables, global variables, and HTML variables in this field. |
| New page for detail block | <p>Check this check box to start a new page in the report for the detail block.</p> |
| Blank lines after block | <p>Specify the number of blank lines after the last detail block line. The range is between 0 and 999.</p> |
| Repeat detail heading | <p>Check this check box to repeat the detail heading before each occurrence of the detail block.</p> |
| Put tabular data at line | <p>Specify where the tabular data line is placed vertically in the detail block. The range is between 1 and 999. If you do not want a tabular data line, specify NONE.</p> |
| Keep block on page | <p>Check this check box to keep the detail block on one page. If you check this check box and a detail block is too long to fit on one page, the detail block is started on a new page. If you do not check this check box, detail blocks can split across two or more pages.</p> |
| Detail block text | <p>You use the buttons above the Detail block text table to add, edit, or remove lines, and also change the lines order.</p> <p>Specify the lines of detail block text.</p> <ul style="list-style-type: none"> • Line: this number specifies where the text is placed vertically in the detail block. You do not have to start with line 1 and you do not have to specify consecutive line numbers. • Alignment: specify where the text is placed horizontally on the line. Valid alignment values are LEFT, RIGHT, CENTER, APPEND, or the number of a specific position in the line. If you specify APPEND, the text is placed immediately following the previous line of text. • Text: type the text that appears in the detail block. You can use form variables, global variables and HTML variables in this field. |

Specifying Options formatting options

You use the **Options** formatting options to define miscellaneous options that control the appearance of the report.

About this task

| <i>Table 30. Options section elements for a classic report</i> | |
|--|---|
| Element | Description |
| Detail spacing | Specify the number of lines for the spacing between tabular data lines or detail blocks. This value must be a number between 1 and 999. |
| Line wrapping width | This field applies only to forms created using the ISPF version of the application QMF for TSO/CICS. Specify the number of characters at which to wrap the columns in the report. The range is between 0 and 999 or the word NONE. |
| Report text line width | For reports, specify the width of the final text, detail heading text, detail block text, and break text in a report. The range is between 1 and 32767, or the words DEFAULT or COLUMNS. For DEFAULT, break footing text and final footing text use the full width of all columns up to the first summary column. For COLUMNS, all text areas use the full width of all columns. |
| Number of fixed columns in report | Specify the number of columns that remain in place when you scroll reports horizontally on the screen or break reports onto multiple pages when printing. The range is between 1 and 999, or the word NONE. |
| Outlining for break columns | Select this check box to display the value in columns with the BREAK usage code when the value changes. |
| Default break text (*) | Select this check box to generate break footing text in breaks for which you did not indicate break footing text. The default break text consists of one asterisk for the highest numbered break level text, two asterisks for the next-highest numbered break level text, and so on. |
| Function name in column heading when grouping | Select this check box to add the name of the summarization function to the heading of the aggregated columns, if a report has summarized columns and you use the GROUP usage code to suppress the tabular data lines. |
| Column wrapped lines kept on a page | This field applies only to forms created using the ISPF version of the application QMF for TSO/CICS. Select this check box to split wrapped columns between two pages, if you specified column wrapping for one or more columns. |
| Across summary column | Select this check box to display the automatically generated ACROSS summary column, which produces additional columns that summarize (total) across the specified columns. |

Table 30. Options section elements for a classic report (continued)

| Element | Description |
|--|--|
| Automatic reordering of report columns | Select this check box to automatically reorder the columns in a report when you specify a BREAK or GROUP usage code, or one of the aggregating usage codes. If selected, the columns are reordered so that BREAK columns are to the far left, GROUP columns are to the left after BREAK columns, all non-aggregated columns are to the left after BREAK and GROUP columns, and all aggregated columns are to the far right. |
| Page renumbering at the highest break level | Select this check box if a printed report should begin a new page beginning with the number 1 whenever the value in the control column of the highest break level changes. Note: The highest break level is the one with the lowest number. |
| Column heading separator | Select this check box to display a row of dashed lines between the column headings and the tabular data lines. |
| Across heading separator | Select this check box to mark columns in ACROSS reports with dashed lines and arrows. |
| Break summary separator | Select this check box to display a row of equal signs between the tabular data lines and the break summary. |
| Final summary separator | Select this check box to display a row of equal signs between the tabular data lines and the final summary. |
| Place LOB contents inline | Select this check box to display LOB contents inline in the contents of the form. |

Specifying Final formatting options

You use the **Final** formatting options to define the content and placement of your report's final text. For example, you can include final text and summary data at the end of the report.

About this task

Table 31. Final section elements for a classic report

| Element | Description |
|----------------------------------|---|
| New page for final text | Check this check box to start a new report page for the final text. |
| Put final summary at line | Specify where the final summary line is placed vertically in the final text lines. The range is between 1 and 999. If you do not want a final summary line, specify NONE. |
| Blank lines before text | Specify the number of blank lines before the first and final text line. The range is between 0 and 999 or the word BOTTOM, which is treated as a zero. |

Table 31. Final section elements for a classic report (continued)

| Element | Description |
|-------------|---|
| Text | <p>You use the buttons above the Final text table to add, edit, or remove lines, and also change the lines order.</p> <p>Specify the lines of final text:</p> <ul style="list-style-type: none"> • Line: specify where the final text is placed vertically. You do not have to start with line 1 and you do not have to specify consecutive line numbers. • Alignment: specify where the text is placed horizontally on the line. Valid alignment values are LEFT, RIGHT, CENTER, APPEND, or the number of a specific position in the line. If you specify APPEND, the text is placed immediately following the previous line of text. • Text: type the text that appears at the bottom of the report. You can use form variables, global variables and HTML variables in this field. |

Specifying Page formatting options

You use the **Page** options to define the content and placement of the page heading and page footing on your report.

About this task

Table 32. Page section elements for a classic report

| Element | Description |
|-----------------------------------|--|
| Blank lines before heading | Specify the number of blank lines before the first page heading line. The range is between 0 and 999. |
| Blank lines after heading | Specify the number of blank lines after the last page heading line. The range is between 0 and 999. |
| Page heading text | <p>You use the buttons above the Page heading text table to add, edit, or remove lines, and also change the lines order.</p> <p>Type each line of page heading text.</p> <ul style="list-style-type: none"> • Line: specify where to place the final text vertically. You do not have to start with line 1 and you do not have to specify consecutive line numbers. • Alignment: specify where to place the text horizontally on the line. Valid alignment values are LEFT, RIGHT, CENTER, APPEND, or the number of a specific position in the line. If you specify APPEND, the text is placed immediately following the previous line of text. • Text: type the final text that appears in the page heading on the report. You can use form variables, global variables and HTML variables in this field. |
| Blank lines before footing | Specify the number of blank lines before the first page footing line. The range is between 0 and 999. |
| Blank lines after footing | Specify the number of blank lines after the last page footing line. The range is between 0 and 999. |

Table 32. Page section elements for a classic report (continued)

| Element | Description |
|--------------------------|--|
| Page footing text | <p>You use the buttons above the Page footing text table to add, edit, or remove lines, and also change the lines order.</p> <p>Specify each line of text printed in the page footing at the bottom of the page:</p> <ul style="list-style-type: none"> • Line: specify where to place the final text vertically. You do not have to start with line 1 and you do not have to specify consecutive line numbers. • Alignment: specify where to place the text horizontally on the line. Valid alignment values are LEFT, RIGHT, CENTER, APPEND, or the number of a specific position in the line. If you specify APPEND, the text is placed immediately following the previous line of text. • Text: type the page footing text that appears at the bottom of the report. |

Using form variables

Form variables are codes that you can insert into text fields to produce information on the report.

For example, you can insert a date variable to produce the current date whenever the report is printed. The following form variables are available:

- **&ROW:** This variable will display the number of the current row of data.
- **&DATE:** This variable will display the current date.
- **&TIME:** This variable will display the current time.
- **&PAGE:** This variable will display the current page number.
- **&COUNT:** This variable will display the number of rows retrieved or printed since the last break at the same level.
- **&CALCid:** This variable identifies a Form calculation expression to use, where "id" is the ID number of the expression.
- **&n:** This variable will display the value of a column, where "n" is the column number.
- **&an:** This variable will display the aggregation of a column, where "n" is the column number and "a" is one of the following aggregation variables: AVG, COUNT, CPCT, CSUM, FIRST, LAST, MAX, MIN, PCT, STDEV, SUM, TCPCT, or TPCT. The aggregation is based on the rows retrieved or printed since the last break at the same level.
- **&Global Variables:** This will display the value of the global variable.
- **&HTML Variables:** This will display the value of the HTML variable.

Different variables are available for use depending on the part of the form that you are editing. The following list shows the types of variables that you can use in each part of a form:

- **Page Heading:** You can use all variables except &an, &COUNT, or &CALCid.
- **Page Footing:** You can use all variables except &an, &COUNT, or &CALCid.
- **Break Heading:** You can use all variables except &an, &COUNT, or &CALCid.
- **Break Footing:** You can use all variables.
- **Calculation Expression:** You can use all variables except &CALCid.
- **Column Expression:** You can use all variables except &Page, &an, &COUNT, or &CALCid.
- **Condition:** You can use all variables except &Page, &an, &COUNT, or &CALCid.

- **Detail Heading** You can use all variables except &COUNT or &CALCid.
- **Detail Block:** You can use all variables.
- **Final Text:** You can use all variables.

Using usage codes

Usage codes provide summary information about the data in a column.

For example, usage codes can provide total summary information at the end of a column, or partial summaries at control breaks in a table. The usage codes available depend on the data in the column and the type of summary.

| Usage Code | Description | Notes |
|------------|--|---|
| ACROSS | Enables you to produce a report with horizontal control breaks. | <p>In an ACROSS report:</p> <ul style="list-style-type: none"> • The number and titles of the columns in the report are dependent on the values in the ACROSS column. There is one set of report columns for each value in the ACROSS column and the header for each is the value of column. The set of report columns includes a column for each one that uses an aggregation usage code (for example, SUM, AVERAGE, COUNT). • You can only have one ACROSS column in a report. • The number of rows and the title of each row in the report are dependent on the values in the GROUP column(s). There is one row for each value in the GROUP column(s) and the title of each row is the value of the column(s). • The CSUM, PCT, CPCT, TPCT and TCPCT usage codes are only partially supported when generating reports that also use the ACROSS usage code. |
| AVERAGE | Average of the values in the column | This usage code is only valid for numeric data This calculated value appears as a total in the report. The calculated value is formatted with the edit code of the column. |
| Breakn | Provide a control break level (where "n" represents a number between 1 and 6). | For example, BREAK1 specifies a control column for a level-1 break and BREAK2 specifies a control column for a level-2 break. Any change in the value of the column causes a section break in the report. Subtotals are displayed for columns whose usage is one of the aggregation usages. Also, the text specified the appropriate Form Break component is displayed. Your query should use an ORDER BY clause that matches your BREAK columns. |
| BREAKnX | Same as BREAKn, except the control column is omitted from the report. | Same as BREAKn. |

| Usage Code | Description | Notes |
|------------|---|---|
| CALCid | The evaluation of the calculation expression in the Form calculations component where ID equals "id". | This calculated value appears as a total in the report and applies only to the last row of data. The calculated value is formatted with the edit code of the column. If the column value is used in the calculation, only the last row of data is evaluated. |
| COUNT | Count of the non-null values in the column. | This calculated value appears as a total in the report. The calculated value is formatted with the edit code K. |
| CPCT | Cumulative percentage each value of the column is of the current total. | <ul style="list-style-type: none"> This calculated value replaces each detail line value and also appears as a total in the report. The calculated value is formatted with the edit code of the column. The CPCT usage code is only partially supported when generating reports that also use the ACROSS usage code. |
| CSUM | Cumulative sum of the values in the column. | <ul style="list-style-type: none"> This calculated value replaces each detail line value and also appears as a total in the report. The calculated value is formatted with the edit code of the column. The CSUM usage code is only partially supported when generating reports that also use the ACROSS usage code. |
| FIRST | First value of the column. | This calculated value appears as a total in the report. The calculated value is formatted with the edit code of the column. |
| GROUP | Displays only one line of summary data for each set of values in the column. | More than one column can have usage code GROUP. If so, a change in value in any column starts a new group. All other columns with no usage code are omitted from the report. |
| LAST | Last value in the column. | This calculated value appears as a total in the report. The calculated value is formatted with the edit code of the column. |
| MAXIMUM | Maximum value in the column. | This calculated value appears as a total in the report. The calculated value is formatted with the edit code of the column. |
| MINIMUM | Minimum value in the column. | This calculated value appears as a total in the report. The calculated value is formatted with the edit code of the column. |
| OMIT | Excludes the column from the report. | The column and its values are not included in the tabular report. The values in the column can still appear in the report (for example, in a break footer) by use of form variables (such as &n, where n represents the column number). |

| Usage Code | Description | Notes |
|------------|---|--|
| PCT | Percentage each value of the column is of the current total. | <ul style="list-style-type: none"> This calculated value replaces each detail line value and also appears as a total in the report. The calculated value is formatted with the edit code of the column. The PCT usage code is only partially supported when generating reports that also use the ACROSS usage code. |
| STDEV | Standard deviation of the values in the column. | This usage code is only valid for numeric data. This calculated value appears as a total in the report. The calculated value is formatted with the edit code of the column. |
| SUM | Sum of the values in the column. | This usage code is only valid for numeric data. This calculated value appears as a total in the report. The calculated value is formatted with the edit code of the column. |
| TPCT | Percentage each value of the column is of the final total. | <ul style="list-style-type: none"> This calculated value replaces each detail line value and also appears as a total in the report. The calculated value is formatted with the edit code of the column. The TPCT usage code is only partially supported when generating reports that also use the ACROSS usage code. |
| TCPCT | Cumulative percentage each value of the column is of the final total. | <ul style="list-style-type: none"> This calculated value replaces each detail line value and also appears as a total in the report. The calculated value is formatted with the edit code of the column. The TCPCT usage code is only partially supported when generating reports that also use the ACROSS usage code. |

Saving classic reports

When you save a classic report you are actually saving the form that is used to create the report. The form is saved as an object and can be saved in a local or network file, in your workspace, in a QMF catalog, or in a repository.

About this task

Saved forms open differently depending on whether the form was saved with an associated data source object.

- When you save the form with an associated data source object then you are saving both elements that are used to create the report. Upon opening a form that has been saved with both the data source object and the form: the data source object is located; the data source object is run and the query results are obtained; the form formatting is applied; and the classic report is presented. Specifying that a particular data source object will always be used when a form is opened streamlines the steps required to create a classic report.
- When you save the form without an associated data source object, then you are only saving the formatting options. When you open the form, you will must supply the query results. That means you

must run a query, make the query results active and open the form in order to create the classic report. If no query results are available then the form opens only for editing.

Saving a form with an associated data source object

When you save the form with an associated data source object then both elements used to create the report are saved. This method of saving streamlines the steps required to create a classic report.

About this task

To save a form with an associated data source object:

Procedure

1. Create a query object and save the query object or open a saved query object.
2. Run the query object to obtain query results.
3. Generate a classic report using a new or saved form; or default form formatting or formatting based on the query results.
4. Save the form. When you open this saved form, it has a known associated data source object (because you saved the object). The data source object is located, the data source object is run and the query results are obtained, the form formatting is applied and the classic report is presented.

Saving a form without an associated data source object

When you save the form without an associated data source object, then you only save the formatting options.

About this task

To save a form without an associated data source object:

1. Create a query object or open an existing query.
2. Run the query object to obtain query results.
3. Generate a classic report using either default settings, formatting settings based on the query results, opening a saved form, or creating a new form.
4. Select **Form > Data Source**. Clear the **Always use this object when opening this document** check box.
5. Save the form. When you open this saved form, it has no known associated data source object. If a set of query results are active, they will be formatted into a classic report, otherwise the form will open for editing.

Working with quick reports

You use quick reports to rapidly generate a printable report from a query result set.

You use the **Create New Quick Report** wizard to define the report 'break' sections, determine the columns over which data will be reported. For example, report sales by region, followed by country within each region. In the **Create New Quick Report** wizard, you can also define the aggregation function applied to each column value (ascending, descending, count, maximum, minimum, sum, average, and so on), as well to define the formatting (text, alignment, font, background color, and so on) of both the detail and summary rows of the report.

Creating quick reports

You can create a quick report from the data in a query by using the **Create New Quick Report** wizard.

About this task

To create a quick report, perform the following procedure:

Procedure

1. To open the **Create New Quick Report** wizard, select **File > New > Other**, expand the **QMF Objects** and select **Quick Report**.
2. On the first page of the **Create New Quick Report** wizard, specify the quick report name and select the required source query. Select the **Copy settings from grid** check box to apply such grid settings as grouping and detail columns distribution and formatting options to a quick report. Click **Next**.
3. On the **Columns** page of the **Create New Quick Report** wizard, select columns for grouping and detail columns. Click **Next**.

Note: You can create a quick report on the basis of an opened query result set. With an active query in the editor window, select **Query > Transfer To > Quick Report**. In this case, the first page of the **Create New Quick Report** wizard is skipped and the **Columns** page is opened automatically.

4. On the **Format** page of the wizard, set the following quick report options:
 - Report title and total line format options
 - Report header and footer format options
 - Columns order
 - Column headers format options
 - Column groups
 - Total and detail columns sets
 - Conditional format for report columns
 - Grouping columns format options
 - Header and Footer of grouping columns
 - Detail columns format options
 - Custom data objects that can be assigned to report columns.
5. Click **Finish**. The quick report is displayed on the **Design** page of the editor where you can manually customize the look of the quick report.

Related tasks

Editing quick reports

You can edit the layout of an existing Quick report by adding, removing, or reordering the grouping and detail columns.

Modifying your quick report design

Once you have created a quick report, you can make simple changes to the layout right in the quick report editor.

Arranging grouping and detail columns

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

Before you begin

You must open the Create New Quick Report wizard.

About this task

To arrange the grouping and detail columns in a quick report:

Procedure

1. Open the **Columns** page of the **Create New Quick Report** wizard.
2. To add or remove grouping columns, click **Move to grouping columns set** or **Remove from grouping columns set**.
3. To add or remove detail columns, click **Move to detail columns set** or **Remove from detail columns set**.
Note: You can add all available columns to the **Detail Columns** list by clicking **Move all to detail columns set**. To remove all columns from the **Detail Columns** list, click **Move all from detail columns set**.
4. To convert a grouping column to detail column and vice versa, select the columns and then click **Move to grouping columns set** or **Move to detail columns set** buttons.
5. To change the order of grouping columns, use **Move selected columns up** or **Move selected columns down** buttons to the right of the **Grouping Columns** list.
6. If you have specified all the required parameters for the quick report, click **Finish** to close the **Create New Quick Report** wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

What to do next

Next: [“Specifying how to display the report title and report total information” on page 267](#)

Related tasks

[“Specifying how to display the report title and report total information” on page 267](#)

You can specify how to display the report title and report total information of a quick report.

[“Specifying how to display column headers” on page 270](#)

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

[“Creating column groups” on page 271](#)

To limit the data that is displayed in a quick report, you can organize the columns in mutually exclusive groups of columns.

[“Creating total and detail column sets” on page 272](#)

You can associate detail columns that you do not always want to be displayed in a quick report with a total column.

[“Applying conditional formatting” on page 273](#)

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

[“Specifying format options for the Grouping Columns ” on page 275](#)

You can specify format options for the Grouping Columns of a quick report.

[“Specifying how to display Header and Footer information” on page 276](#)

You can specify the how to display Header and Footer information for a grouping column of a quick report.

[“Specifying format options for the Details Columns” on page 278](#)

You can specify how to display the Details Columns of a quick report.

[“Specifying quick report column format options” on page 279](#)

You can specify the formatting options of individual columns of your quick report.

[“Assigning custom data objects to quick report columns” on page 281](#)

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Specifying how to display the report title and report total information

You can specify how to display the report title and report total information of a quick report.

Before you begin

You must open the Create New Quick Report wizard.

About this task

To specify how to display the report title and report total information in the quick report:

Procedure

1. Open the **Format** page of the **Create New Quick Report** wizard.
2. Select **Report Settings** from the tree.
3. Select the **Visible** check box in the **Report Title** section to display the title of the report.
The **Report Title** options become available.
4. Specify the title of the report in the **Text** field.
Note: You can view the quick report title only on the printed report.
5. From the **Alignment** menu, specify where on the page the report title must be displayed.
Valid options are:
 - **Left** - The title is displayed on the left hand side of the header.
 - **Center** - The title is displayed in the center of the header.
 - **Right** - The title is displayed on the right hand side of the header.
6. Specify the font of the title in the **Font** field.
7. Specify the background color of the title in the **Background** field.
8. Select the **Visible** check box in the **Total Line** section to include a totals line at the bottom of the report.
The **Total Line** options become available.
9. Specify the font of the totals line text in the **Font** field.
10. Specify the background color of the totals line in the **Background** field.
11. In the **Common settings** area, specify how to display the quick report grid:
 - To display row and column numbers, select the **Show row and column numbers** check box.
 - To display grid lines, select the **Show grid lines** check box.By default, both check boxes are cleared.
12. If you have specified all the required parameters for the quick report, click **Finish** to close the **Create New Quick Report** wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

What to do next

Next: [“Specifying how to display column headers” on page 270](#)

Related tasks

[“Arranging grouping and detail columns” on page 265](#)

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

[“Specifying how to display column headers” on page 270](#)

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

[“Creating column groups” on page 271](#)

To limit the data that is displayed in a quick report, you can organize the columns in mutually exclusive groups of columns.

[“Creating total and detail column sets” on page 272](#)

You can associate detail columns that you do not always want to be displayed in a quick report with a total column.

[“Applying conditional formatting” on page 273](#)

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

[“Specifying format options for the Grouping Columns ” on page 275](#)

You can specify format options for the Grouping Columns of a quick report.

[“Specifying how to display Header and Footer information” on page 276](#)

You can specify the how to display Header and Footer information for a grouping column of a quick report.

[“Specifying format options for the Details Columns” on page 278](#)

You can specify how to display the Details Columns of a quick report.

[“Specifying quick report column format options” on page 279](#)

You can specify the formatting options of individual columns of your quick report.

[“Assigning custom data objects to quick report columns” on page 281](#)

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Formatting report header and footer

Before you begin

You must open the Create New Quick Report wizard.

About this task

The **Report Header** and **Report Footer** allows you to format the text in the header and footer for PDF export.

You can format and style header and footer using the following options:

- Edit the content using multiline text
- Clipboard operations such as cut, copy, paste (only available in QMF for Workstation)
- Redo and undo
- Search and replace (only available in QMF for Workstation)
- Formatting operations such as lists, bullets, horizontal indentation, text and image alignment, copy and remove format
- Font and paragraph (only available in QMF for Workstation) formatting
- Color Text and background using color selection pallet (only available in QMF for Workstation)
- Source edit, which replaces the current text editor in a HTML friendly way
- Insert table (only available in QMF for Workstation), horizontal line, image. You can insert image using the image URL. We recommend that you use the local folder path instead of the web URL.
- Show blocks, which displays the structure of your page as outline (only available in QMF for Workstation)
- Code to insert specific data in the exported PDF. You can use the following codes:

| Code | Description |
|------|-----------------------------------|
| &f | Prints the name of the object. |
| &n | Prints the current page number. |
| &N | Prints the total number of pages. |
| &d | Prints the current date. |
| &t | Prints the current time. |
| && | Prints a single ampersand. |

For example:

If you enter the code Footer &n → curr &N → Total page &!1,2! in the footer, where:

- &n will be replaced the current page number
- &N will be replaced total number of pages
- &!1, 2! denotes the cell numbers of the table

Then the report when exported to generates a PDF with the following footer text:

Footer 1 → cure 1 → Total page Happy

Usability notes:

- Quick header and footer formatting overrides page head and page footer formatting.
- If you want to reduce the line spacing, press **Shift + Enter**.
- If you want to reduce the space after horizontal line, drag the horizontal rule downwards.
- If you want to save the modified text, press **Ctrl + S**.

Procedure

1. Open the Format page of the **Create New Quick Report** wizard.
2. Expand **Report Settings** node and select **Report Header** or **Report Footer** node.
3. On the **Report Header** or **Report Footer** page, type in the text in the **Enter Header Text** or **Enter Footer Text** pane.
4. Format the text using the **Format** options available. You can also pass on an expression.
5. Click **Finish**.
6. Export the report to verify your header and footer formatting changes.

Related tasks

[“Arranging grouping and detail columns” on page 265](#)

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

[“Specifying how to display the report title and report total information” on page 267](#)

You can specify how to display the report title and report total information of a quick report.

[“Specifying how to display column headers” on page 270](#)

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

[“Creating column groups” on page 271](#)

To limit the data that is displayed in a quick report, you can organize the columns in mutually exclusive groups of columns.

[“Applying conditional formatting” on page 273](#)

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

[“Specifying format options for the Grouping Columns” on page 275](#)

You can specify format options for the Grouping Columns of a quick report.

[“Specifying how to display Header and Footer information” on page 276](#)

You can specify the how to display Header and Footer information for a grouping column of a quick report.

[“Specifying format options for the Details Columns” on page 278](#)

You can specify how to display the Details Columns of a quick report.

[“Specifying quick report column format options” on page 279](#)

You can specify the formatting options of individual columns of your quick report.

[“Assigning custom data objects to quick report columns” on page 281](#)

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Specifying how to display column headers

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

Before you begin

You must open the Create New Quick Report wizard.

About this task

To specify how to display column headers:

Procedure

1. Open the **Format** page of the **Create New Quick Report** wizard.
2. Expand **Report Settings** node and select **Column Headers** node from the tree.
3. To change the order of the columns, select a column and drag it to the required position.
4. To hide a column, select this column and then select the **Hide** check box.
5. To create hierarchical column headers or group existing column headers, click **Create Header**, and drag the required column headers to this header.

Note: If you create a header and do not populate it with at least one of existing headers, it will not be displayed in the report.

6. In the **Header** area, specify the name, font, background color, horizontal and vertical alignment of the column header.
7. If you have specified all the required parameters for the quick report, click **Finish** to close the **Create New Quick Report** wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

What to do next

Next: [“Creating column groups” on page 271](#)

Related tasks

[“Arranging grouping and detail columns” on page 265](#)

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

[“Specifying how to display the report title and report total information” on page 267](#)

You can specify how to display the report title and report total information of a quick report.

[“Creating column groups” on page 271](#)

To limit the data that is displayed in a quick report, you can organize the columns in mutually exclusive groups of columns.

[“Creating total and detail column sets” on page 272](#)

You can associate detail columns that you do not always want to be displayed in a quick report with a total column.

[“Applying conditional formatting” on page 273](#)

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

[“Specifying format options for the Grouping Columns ” on page 275](#)

You can specify format options for the Grouping Columns of a quick report.

[“Specifying how to display Header and Footer information” on page 276](#)

You can specify the how to display Header and Footer information for a grouping column of a quick report.

[“Specifying format options for the Details Columns” on page 278](#)

You can specify how to display the Details Columns of a quick report.

[“Specifying quick report column format options” on page 279](#)

You can specify the formatting options of individual columns of your quick report.

[“Assigning custom data objects to quick report columns” on page 281](#)

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Creating column groups

To limit the data that is displayed in a quick report, you can organize the columns in mutually exclusive groups of columns.

Before you begin

You must open the Create New Quick Report wizard.

About this task

You can then toggle between the groups to display only those columns that are assigned to that group. Columns assigned to other groups are not displayed. Any columns not assigned to any group are always displayed in the quick report.

To create a column group:

Procedure

1. Open the **Format** page of the **Create New Quick Report** wizard.
2. Expand **Report Settings** node and select **Column Groups** node from the tree.
3. On the **Column Groups** page, click **Add group**. A new group is displayed in the **Groups** list.
4. Add the required columns to the group by selecting them from the **Columns** list.
5. To hide the columns already assigned to a group from the **Columns** list, clear the **Show grouped columns** check box.
6. To change the order of the column groups, select a group and click **Move Up** or **Move Down**.
Note: The first group in the list is displayed by default when you open the quick report.
7. If you have specified all the required parameters for the quick report, click **Finish** to close the **Create New Quick Report** wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

What to do next

Next: [“Creating total and detail column sets” on page 272](#)

Related tasks

[“Arranging grouping and detail columns” on page 265](#)

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

[“Specifying how to display the report title and report total information” on page 267](#)

You can specify how to display the report title and report total information of a quick report.

[“Specifying how to display column headers” on page 270](#)

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

[“Creating total and detail column sets” on page 272](#)

You can associate detail columns that you do not always want to be displayed in a quick report with a total column.

[“Applying conditional formatting” on page 273](#)

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

[“Specifying format options for the Grouping Columns ” on page 275](#)

You can specify format options for the Grouping Columns of a quick report.

[“Specifying how to display Header and Footer information” on page 276](#)

You can specify the how to display Header and Footer information for a grouping column of a quick report.

[“Specifying format options for the Details Columns” on page 278](#)

You can specify how to display the Details Columns of a quick report.

[“Specifying quick report column format options” on page 279](#)

You can specify the formatting options of individual columns of your quick report.

[“Assigning custom data objects to quick report columns” on page 281](#)

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Creating total and detail column sets

You can associate detail columns that you do not always want to be displayed in a quick report with a total column.

Before you begin

You must open the Create New Quick Report wizard.

About this task

For example, you might have a *Product Costs* column that is calculated from the *Item Cost* and *Number of Items Shipped*. In the quick report you might want only the *Product Costs* column to be displayed, but you still want the other columns available if needed. You can use this feature to make the *Product Costs* column the total column, and specify the columns used to calculate the total column as detail columns.

You can display or hide detail columns by clicking the arrow in the total column header.

To create total and detail column set:

Procedure

1. Open the Format page of the **Create New Quick Report** wizard.
2. Expand **Report Settings** node and select **Total/Detail Sets** node from the tree.
3. On the **Total/Detail Sets** page, click **Add Total/Detail Set**. A new set is displayed in the table.
4. Specify a total column from the list of available columns. All of the non-selected columns are automatically displayed in the **Detail Columns** list.
5. From the **Detail Columns** list, select detail columns to associate them with the specified total column

6. If you have specified all the required parameters for the quick report, click **Finish** to close the **Create New Quick Report** wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

What to do next

Next: [“Applying conditional formatting” on page 273](#)

Related tasks

[“Arranging grouping and detail columns” on page 265](#)

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

[“Specifying how to display the report title and report total information” on page 267](#)

You can specify how to display the report title and report total information of a quick report.

[“Specifying how to display column headers” on page 270](#)

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

[“Creating column groups” on page 271](#)

To limit the data that is displayed in a quick report, you can organize the columns in mutually exclusive groups of columns.

[“Applying conditional formatting” on page 273](#)

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

[“Specifying format options for the Grouping Columns ” on page 275](#)

You can specify format options for the Grouping Columns of a quick report.

[“Specifying how to display Header and Footer information” on page 276](#)

You can specify the how to display Header and Footer information for a grouping column of a quick report.

[“Specifying format options for the Details Columns” on page 278](#)

You can specify how to display the Details Columns of a quick report.

[“Specifying quick report column format options” on page 279](#)

You can specify the formatting options of individual columns of your quick report.

[“Assigning custom data objects to quick report columns” on page 281](#)

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Applying conditional formatting

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

Before you begin

You must open the **Create New Quick Report** wizard.

About this task

Conditional format allows you to specify conditional expressions that control both the display of the data and the behavior of a quick report based on the underlying data. For example, you can use conditional format to highlight in red any fields that indicate year-end sales that fall below a particular figure.

To apply conditional formatting:

Procedure

1. Open the Format page of the **Create New Quick Report** wizard.
2. Expand **Report Settings** node and select **Conditional Format** node from the tree.

3. On the **Conditional Format** page, click **Add Format**.
4. On the first page of the **New Conditional Format** wizard, specify the conditional format name and select the columns and levels of detail to which this format will be applied. Click **Next**.

Note: If the format for the selected columns or levels of detail has already been set, the new format will override it.
5. On the second page of the wizard, specify conditional expressions and the format options.
6. From the **Base column** list, select the column which values will be used as a part of condition for formatting. You can select the **<Based on the same column>** option, to use the values from the column to which you apply the conditional format. If you select this option, the format can only be applied to the columns of the same type (numeric, string, or date/time).
7. Click **Add Condition** and specify condition parameters in the **Selected condition** area.
 - a) From the **Operator** list, select the required operator. Available options are:
 - **Not Less Than** - All values that are equal to or greater than the specified value are included.
 - **Greater Than** - All values that are greater than the specified value are included.
 - **Not Greater Than** - All values that are equal to or less than the specified value are included.
 - **Less Than** - All values that are less than the specified value are included.
 - **Equal To** - Only values that are equal to the specified value are included.
 - **Not Equal To** - All values that are not equal to the specified value are included.
 - **Like** - All values that match the specified pattern are included. The pattern can contain the following special characters:
 - % matches zero (0) or more characters.
 - _ matches only one (1) character.
 - ? matches only one (1) character.
 - **Case-Insensitive Like** - A case-insensitive version of the Like operator.
 - **Unlike** - All values that are not like the specified value are included.
 - **Is Null** - Only null values are included.
 - **Is Not Null** - All of the values that are not null are included.
 - **In Range** - All of the values that fall within the two specified values, along with the values themselves, are included.
 - **Outside Range** - All of the values that fall outside of the two values, along with the values themselves, are included.
 - **Between** - All of the values that fall within the two specified values are included.
 - **Not Between** - All of the values that fall outside of the two specified values are included.
 - b) Specify the types and values for the operands. You can set either a constant value or a name of the column that will provide required values.
8. In the editor, specify the conditional format options. You can preview the resulting format in the **Sample** area.
 - a) On the **Text** page of the editor, set the text font, data format, and alignment.
 - b) On the **Colors** page, set the foreground and background colors. You can use either a single color or gradient.
 - c) On the **Image** page, you can select an image to display in the report and set its position. Available position options are:
 - Before text
 - After text
 - Replace text.
9. Click **Finish** to close the **New Conditional Format** wizard.

The created condition format is displayed on the **Conditional Format** page of the **Create New Quick Report** wizard.

10. If you have specified all the required parameters for the quick report, click **Finish** to close the **Create New Quick Report** wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

Results

You can edit created conditional formats by using the **Edit Conditional Format** wizard. You can also reassign created conditional formats to other columns or levels of detail on the **Assignments** page.

What to do next

Next: [“Specifying format options for the Grouping Columns ” on page 275](#)

Related tasks

[“Arranging grouping and detail columns” on page 265](#)

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

[“Specifying how to display the report title and report total information” on page 267](#)

You can specify how to display the report title and report total information of a quick report.

[“Specifying how to display column headers” on page 270](#)

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

[“Creating total and detail column sets” on page 272](#)

You can associate detail columns that you do not always want to be displayed in a quick report with a total column.

[“Specifying format options for the Grouping Columns ” on page 275](#)

You can specify format options for the Grouping Columns of a quick report.

[“Specifying how to display Header and Footer information” on page 276](#)

You can specify the how to display Header and Footer information for a grouping column of a quick report.

[“Specifying format options for the Details Columns” on page 278](#)

You can specify how to display the Details Columns of a quick report.

[“Specifying quick report column format options” on page 279](#)

You can specify the formatting options of individual columns of your quick report.

[“Assigning custom data objects to quick report columns” on page 281](#)

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Specifying format options for the Grouping Columns

You can specify format options for the Grouping Columns of a quick report.

Before you begin

You must open the Create New Quick Report wizard.

About this task

To specify **Grouping Columns** formatting options:

Procedure

1. Open the **Format** page of the **Create New Quick Report** wizard.
2. Select **Grouping Columns** from the tree.
3. To change a grouping column's name, click the **Header** cell and specify the new text.

The text that you specify will display as the grouping column name in the quick report. To set a multiline header you should use “_” symbol to separate lines. In the report it will be substituted by the line break character.

4. To change the sorting order of a grouping column, click the **Sort** cell and select **Ascending** or **Descending**.
5. To change the width of a grouping column, click the **Width** cell and specify the new width.
6. If you have specified all the required parameters for the quick report, click **Finish** to close the **Create New Quick Report** wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

What to do next

Next: [“Specifying how to display Header and Footer information” on page 276](#)

Related tasks

[“Arranging grouping and detail columns” on page 265](#)

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

[“Specifying how to display the report title and report total information” on page 267](#)

You can specify how to display the report title and report total information of a quick report.

[“Specifying how to display column headers” on page 270](#)

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

[“Applying conditional formatting” on page 273](#)

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

[“Specifying how to display Header and Footer information” on page 276](#)

You can specify the how to display Header and Footer information for a grouping column of a quick report.

[“Specifying format options for the Details Columns” on page 278](#)

You can specify how to display the Details Columns of a quick report.

[“Specifying quick report column format options” on page 279](#)

You can specify the formatting options of individual columns of your quick report.

[“Assigning custom data objects to quick report columns” on page 281](#)

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Specifying how to display Header and Footer information

You can specify the how to display Header and Footer information for a grouping column of a quick report.

Before you begin

You must open the Create New Quick Report wizard.

About this task

To specify Header and Footer information for the **Grouping Columns** options:

Procedure

1. Open the **Format** page of the **Create New Quick Report** wizard.
2. In the tree, expand **Grouping Columns**, then one of the column names under it, and select **Header**.
3. To display a header for every grouping column row, select the **Visible** check box.
4. Specify the text that you want to display in the grouping column row header in the **Text** field.
The default text is = 'column_name: ' +@[column] where 'column_name' is the column name of the grouping column and 'column' is the value of the given grouping column row. For example, if your

grouping column is named DEPT and it contains department numbers of a given company, this text could generate sequential row headers of DEPT: 10, DEPT: 20, and so on. You can change this text to anything that you want, but it is recommended that you keep this overall structure.

5. Specify the font of the grouping column row headers in the **Font** field.
6. Specify the background color of the grouping column row headers in the **Background** field.
7. Specify the height of the header in the **Height** field.
8. You can display charts in the Header of a grouping column. To display a chart in the header, select **Enable chart** check box. In the **New chart** wizard specify parameters of the chart.

Note: If you clear the **Enable chart** check box, the specified chart will not be displayed but it is not removed from the quick report.

9. You can edit a created chart in the **Edit chart properties** window. To open the **Edit chart properties** window, click **Chart settings**.
10. To remove a chart, click **Delete chart**.
11. The **Footer** options are identical to the **Header** options. To open the **Footer** page, select **Footer** in the tree.
12. If you want to display the aggregate totals of the specified detail columns, select the **Visible** check box and specify the totals line text in the **Text** field.

The default text is = 'Summary of ' + @[column] where 'column' is the name of a details column which values should be aggregated.
13. If you have specified all the required parameters for the quick report, click **Finish** to close the **Create New Quick Report** wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

What to do next

Next: [“Specifying format options for the Details Columns” on page 278](#)

Related tasks

[“Creating charts” on page 313](#)

You can create a chart from a query result set with the help of the **New Chart** wizard.

[“Arranging grouping and detail columns” on page 265](#)

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

[“Specifying how to display the report title and report total information” on page 267](#)

You can specify how to display the report title and report total information of a quick report.

[“Specifying how to display column headers” on page 270](#)

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

[“Creating column groups” on page 271](#)

To limit the data that is displayed in a quick report, you can organize the columns in mutually exclusive groups of columns.

[“Creating total and detail column sets” on page 272](#)

You can associate detail columns that you do not always want to be displayed in a quick report with a total column.

[“Applying conditional formatting” on page 273](#)

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

[“Specifying format options for the Grouping Columns ” on page 275](#)

You can specify format options for the Grouping Columns of a quick report.

[“Specifying format options for the Details Columns” on page 278](#)

You can specify how to display the Details Columns of a quick report.

[“Specifying quick report column format options” on page 279](#)

You can specify the formatting options of individual columns of your quick report.

[“Assigning custom data objects to quick report columns” on page 281](#)

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Specifying format options for the Details Columns

You can specify how to display the Details Columns of a quick report.

Before you begin

You must open the Create New Quick Report wizard.

About this task

To specify **Details Columns** formatting options:

Procedure

1. Open the **Format** page of the **Create New Quick Report** wizard.
2. Select **Details Columns** from the tree.
The **Details Columns** page opens.
3. To change a detail column's column name, click the **Header** cell and specify the new text.
The text that you specify will display as the detail column name in the quick report. To set a multiline header you should use “_” symbol to separate lines. In the report it will be substituted by the line break character.
4. To change how a detail column is aggregated, click the **Aggregate** cell and select a new value.
Valid options include:
 - Count
 - First
 - Last
 - Maximum
 - Minimum
 - Sum
 - Average
 - Standard DeviationSelect **None** to remove aggregation from a column.
5. To change the width of a detail column, click the **Width** cell and specify the new width.
6. If you have specified all the required parameters for the quick report, click **Finish** to close the **Create New Quick Report** wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

What to do next

Next: [“Specifying quick report column format options” on page 279](#)

Related tasks

[“Arranging grouping and detail columns” on page 265](#)

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

[“Specifying how to display the report title and report total information” on page 267](#)

You can specify how to display the report title and report total information of a quick report.

[“Specifying how to display column headers” on page 270](#)

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

[“Creating column groups” on page 271](#)

To limit the data that is displayed in a quick report, you can organize the columns in mutually exclusive groups of columns.

[“Creating total and detail column sets” on page 272](#)

You can associate detail columns that you do not always want to be displayed in a quick report with a total column.

[“Applying conditional formatting” on page 273](#)

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

[“Specifying format options for the Grouping Columns ” on page 275](#)

You can specify format options for the Grouping Columns of a quick report.

[“Specifying how to display Header and Footer information” on page 276](#)

You can specify the how to display Header and Footer information for a grouping column of a quick report.

[“Specifying quick report column format options” on page 279](#)

You can specify the formatting options of individual columns of your quick report.

[“Assigning custom data objects to quick report columns” on page 281](#)

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Specifying quick report column format options

You can specify the formatting options of individual columns of your quick report.

About this task

In addition to customizing the look and feel of a quick report's summary rows, you can specify the formatting options for each column that is displayed in your report.

To specify individual column formatting options:

Procedure

1. Open the **Format** page of the **Create New Quick Report** wizard.
2. Select a column name from the tree.
The formatting options for each row are identical, whether the column is a grouping column or a detail column.
3. Specify the width of the column in the **Column Width** field.
4. In the **Details** area, specify the font, data format, background color, horizontal and vertical alignment of the column. For the columns containing character data, you can apply automatic text wrapping. To do this, select the **Wrap text** check box.
5. If you have specified all the required parameters for the quick report, click **Finish** to close the **Create New Quick Report** wizard. The quick report is displayed on the **Design** page of the editor where you can manually customize its look.

Related tasks

[“Arranging grouping and detail columns” on page 265](#)

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

[“Specifying how to display the report title and report total information” on page 267](#)

You can specify how to display the report title and report total information of a quick report.

[“Specifying how to display column headers” on page 270](#)

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

[“Creating column groups” on page 271](#)

To limit the data that is displayed in a quick report, you can organize the columns in mutually exclusive groups of columns.

[“Creating total and detail column sets” on page 272](#)

You can associate detail columns that you do not always want to be displayed in a quick report with a total column.

[“Applying conditional formatting” on page 273](#)

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

[“Specifying format options for the Grouping Columns ” on page 275](#)

You can specify format options for the Grouping Columns of a quick report.

[“Specifying how to display Header and Footer information” on page 276](#)

You can specify the how to display Header and Footer information for a grouping column of a quick report.

[“Specifying format options for the Details Columns” on page 278](#)

You can specify how to display the Details Columns of a quick report.

[“Assigning custom data objects to quick report columns” on page 281](#)

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Specifying data format options for quick report columns

You can specify data values format options for individual columns in the **Choose a format** window.

About this task

To specify data format options:

Procedure

1. Open the **Format** page of the **Create New Quick Report** wizard.
2. Select a column name from the tree. In the **Details** area, click the ellipsis (...) next to the **Format** field.
3. Select the format of the data values from the **Format** list. Valid options include:

The following formats are available for numeric data type:

- **As is** - The exact value of the column is displayed without any formatting.
- **Currency** - Column values are displayed as currency. The locale settings of the current session determine the currency unit that is displayed.
- **Decimal** - Column values are displayed as decimals.
- **Scientific** - Column values are displayed in scientific notation.
- **Percent** - Column values are displayed as percentages.
- **Custom** - Data point values are displayed according to the pattern that you specified in the **Pattern** field.

The following formats are available for date data type:

- **As is** - The exact value of the column is displayed without any formatting.
- **Database** - The date is displayed in the format of the database.
- **Date** - Select the date presentation from the available formats.
- **Custom** - The date is displayed according to the pattern that you specified in the **Pattern** field.

The following formats are available for timestamp data type:

- **As is** - The exact value of the column is displayed without any formatting.

- **Database** - The timestamp is displayed in the format of the database.
 - **Timestamp** - Select the timestamp presentation from the available formats.
 - **Date** - Only the date component of the timestamp is displayed.
 - **Time** - Only the time component of the timestamp is displayed.
 - **Custom** - The timestamp is displayed according to the pattern that you specified in the **Pattern** field.
4. To include thousands separators with data point values, select the **Thousands separator** check box.
 5. Specify the number of decimal places that are displayed in the **Decimal slaces** field.
A value of '0' indicates that no decimal places are displayed.
 6. Select the format of negative numbers in the **Negative numbers** list.
 7. Click **OK**.
The **Choose a Format** window closes and control returns to the **Create New Quick Report** wizard.

Assigning custom data objects to quick report columns

You can assign custom data objects that you want to be opened when you click a column in your quick report.

Before you begin

You must open the **Create New Quick Report** wizard.

About this task

To specify click events for a column:

Procedure

1. Open the **Format** page of the **Create New Quick Report** wizard.
2. Expand either a grouping or detail column node and select **Click Events** node from the tree. The **Click Events** page of the **Create New Quick Report** wizard contains two separate pages:
 - **Drill objects** page on which you can assign objects to the selected column.
 - **Column prompts** page on which you can specify variable values that must be passed to the assigned object.
3. On the **Drill objects** page, click **Add new drill object** and browse to the required object in a repository or file system. If the selected object contains prompts, they are displayed in the **Prompts Assignments** list.
Note: Several objects can be assigned to one and the same column. If you click the column, the first object in the list of assigned objects will be opened. To change the order of the objects by clicking **Move Object Up** or **Move Object Down**.
4. You can specify column parameters that will be passed to the assigned object on the **Column prompts** page.
5. To add a column prompt, click **Add new column prompt** and specify its parameters including column prompt name, type, and value. Available type options are:
 - **Constant** to use a constant value
 - **Column** to set the column that will provide the value
 - **Input prompt** to specify the prompt of the query on which the report is based that will pass the value to the drill object you assign to the selected column.
6. Assign the created column prompts with the required prompts of the data object in the **Prompts Assignments** area on the **Drill objects** page. You can try to assign prompts automatically by clicking **Automatch columns by name**. If the names of a column prompt and an object prompt do not coincide, you can assign them manually.

7. If you have specified all the required parameters for the quick report, click **Finish** to close the **Create New Quick Report** wizard and display the quick report on the **Design** page of the editor where you can manually customize the look of the report.

Related tasks

[“Arranging grouping and detail columns” on page 265](#)

You can add and remove the grouping and detail columns from a quick report and convert a grouping column to a detail column and vice versa.

[“Specifying how to display the report title and report total information” on page 267](#)

You can specify how to display the report title and report total information of a quick report.

[“Specifying how to display column headers” on page 270](#)

You can specify the look of column headers and the order in which they are displayed in a quick report on the **Columns Headers** page of the **Create New Quick Report** wizard.

[“Creating total and detail column sets” on page 272](#)

You can associate detail columns that you do not always want to be displayed in a quick report with a total column.

[“Applying conditional formatting” on page 273](#)

You can use a conditional format in quick reports to help you visually explore and analyze data, discover critical issues, and identify patterns and trends.

[“Specifying format options for the Grouping Columns ” on page 275](#)

You can specify format options for the Grouping Columns of a quick report.

[“Specifying how to display Header and Footer information” on page 276](#)

You can specify the how to display Header and Footer information for a grouping column of a quick report.

[“Specifying format options for the Details Columns” on page 278](#)

You can specify how to display the Details Columns of a quick report.


[“Specifying quick report column format options” on page 279](#)


You can specify the formatting options of individual columns of your quick report.

Editing quick reports

You can edit the layout of an existing Quick report by adding, removing, or reordering the grouping and detail columns.

About this task

You can add, remove or reorder the grouping and detail columns in the report on the **Columns** page of the **Edit Quick Report** wizard. To open the **Columns** page of the wizard, click the **Edit Quick Report Columns** toolbar button (.

You can change the report title and totals line, as well as the formatting options of grouping and detail columns on the Format page of the Edit Quick Report wizard. To open the Format page of the wizard, click the Edit Quick Report Format toolbar button(.

These pages are similar to the Column and Format pages of the **Create New Quick Report** wizard.

You can also modify and change the data source you use to create the quick report.

Related tasks

[Creating quick reports](#)

You can create a quick report from the data in a query by using the **Create New Quick Report** wizard.

[Modifying your quick report design](#)

Once you have created a quick report, you can make simple changes to the layout right in the quick report editor.

Changing data source

You can change data sources for already created quick report.

Before you begin

You must create a quick report.

About this task

To change data source:

Procedure

1. With a quick report opened in the editor, select **Report > Change data source**.
2. On the first page of **Edit Quick Report** wizard, specify a new data source. You can select either a previously saved data source or an opened one.
3. If columns in the new data source match the columns of the previously specified data source, you can click **Finish** to close the wizard and update the report. Otherwise, click **Next** to proceed.
4. On the second page of the wizard, match the data source columns with the columns of the report. If there is no required match to a report column, select the **Remove unmatched columns** check box to proceed. All such columns will be excluded from the report.
5. Click **Finish** to close the wizard and update the report.

Note: All report columns remain their initial properties. If after changing data source the column output format does not match the column data type, the default format for this data type is used.

Related tasks

[“Editing data source” on page 283](#)

You can edit related data sources information for already created quick report.

Editing data source

You can edit related data sources information for already created quick report.

Before you begin

You must create a quick report.

About this task

To edit data source:

Procedure

1. With a quick report opened in the editor, select **Report > Edit data source**. The specified data source opens in a separate editor.
2. Edit and save the data source. When you save the modified data source, its columns are compared with the columns of the report. If there is no full compliance, you must match the columns manually in the **Edit Quick Report** wizard.
3. On the **Columns** page of the **Edit Quick Report** wizard, match the data source columns with the columns of the report. If there is no required match to a report column, select the **Remove unmatched columns** check box to proceed. All unmatched columns will be excluded from the report.

Note: If it is impossible to match the report columns and data source columns, close the **Edit Quick Report** wizard and edit the data source. If you do not match the columns, data source will not be saved within the report.

4. Click **Finish** to close the wizard and update the report.

Note: All report columns remain their initial properties. If after editing the column output format does not match the column data type, the default format for this data type is used.

Related tasks

[“Changing data source” on page 283](#)

You can change data sources for already created quick report.

Modifying your quick report design

Once you have created a quick report, you can make simple changes to the layout right in the quick report editor.

Before you begin

You must create a quick report.

About this task

To modify a quick report :

Procedure

1. To expand or collapse the content of a group, click the plus or minus sign to the left of the name of this group.
2. To collapse all groups and show only aggregated information, click the minus sign in the header of the grouping column.
3. To view or hide the contents of detail columns assigned to a total column, click the arrow in the header of the total column.
4. To change the columns order, you can select a column and drag it to another position. The red vertical line indicates the possible column position.
5. To change the width of a column or the height of a row, you can drag the column or row border.
6. To move labels and charts within the header and footer of grouping columns, simply drag them to a new position.

Note: You can also make changes to the size and position of components in the **Properties** view.

7. To add an extra text field to the header or footer of grouping columns, right-click the header or footer and select **Add Text to Break**.
8. To modify the titles displayed in the header or footer and the text fields that you add, use the **Properties** view.
9. To add a chart to the header or footer of grouping columns, right-click the header or footer and select **Add Chart**.
10. To quickly change the sorting of grouping or detail columns, hide selected columns, or add calculated columns, use the column pop-up menu.

Related tasks

[Creating quick reports](#)

You can create a quick report from the data in a query by using the **Create New Quick Report** wizard.

[Editing quick reports](#)

You can edit the layout of an existing Quick report by adding, removing, or reordering the grouping and detail columns.

Form menu

In QMF, the **Form** menu becomes available when a form is the active object in the editor.

The following menu commands are available:

Data Source Object

Opens the **Data Source Object** dialog where you can specify the data source object that will be associated with the form. The data source object for a form is the object that is run to obtain the query results that are formatted into a report.

Convert to Visual Report

Converts the active form into a visual report. The generated visual report opens in a new window.

Convert to HTML

Opens the **Convert to HTML** dialog, which contains the following options:

- Specify which tags will be added to the form in **The following tags will be added to the form** list box.
- Check the **Include tabular data as HTML table** check box to insert any table data into a HTML formatted table.
- Check the **Adjust additional form values for HTML table tags** check box to have the application automatically adjust form values for HTML formatting.
- Check the **Save as user defaults** check box to save the current settings and selections as user defaults.
- Click the **User Defaults** button to reset to user defaults.
- Click the **Product Defaults** button to reset to product defaults.
- Click the **Clear Values** button to clear all selected values from the **Value** column of the list box.

Set Data Source

Select a data source (from a list of available data sources in your repository) where you want the active query to run. The **Set Data Source** dialog opens.

Set User Information

Opens the **Set User Information** dialog where you specify the user ID and password that QMF will use to connect to the data source.

Set Font

Opens the **Font** dialog where you can change the display font attributes of the current classic report in the active window.

Chapter 12. Working with forecasts

Forecasts allow you to predict future values of time series historical data.

With forecasting capabilities, you can make projections of future values based on past values. Using forecasts, organizations can prepare for changes in economic or competitive conditions by analyzing time series historical data to predict performance and future trends. For example, in a supply chain, if the forecast demand matches the actual demand then significant efficiencies can be achieved in terms of production, distribution, and return.

QMF forecasts use various predictive methods based on mathematical algorithms that model the future demand based on time series historical data that can be sourced from queries and tables containing date or time columns. The overall objective is to choose a time series method that produces a best fit model of past values, by identifying existing patterns in the data and projecting the model into the future to generate the forecast.

The following methods can be used to forecast future values:

- If the time series is relatively stationary with no overall tendency to fluctuate at one part of the series as compared to another part of the series, then **Moving Average**, **Weighted Moving Average**, or **Single Exponential Smoothing** provide the best fit model.
- If the time series has a trend with a consistent upward or downward movement over time, then **Double Exponential Smoothing** provide the best fit model.
- If the series has a trend and seasonality with a pattern of peaks and troughs that repeat themselves over a time-frame of usually less than or equal to a year, then **Holt-Winters method** provide the best fit model.
- If the series has a trend, seasonality and cyclicity with a pattern of peaks and troughs that repeat themselves over an extended time-frame usually greater than a year, then the **Multiplicative Decomposition method** provide the best fit model.
- If the series displays none of the aforementioned methods, then **Neural Networks** will be used to mathematically fit the historical data.
- If there are theoretical reasons to indicate that the data should follow a clear mathematical function, then one of the **Curve fitting methods** can be used.
- In addition to the aforementioned methods, the forecaster is also able to manually adjust any predicted values based on the forecaster's knowledge and any external events.

As most new users discover, the ability to quickly plot and compare each forecast method is a major feature of QMF forecasts. However, a forecaster's knowledge and experience help to reduce the possibilities and consequently provide greater confidence and reliability in the forecast.

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

About this task

To create a forecast, perform the following procedure:

Procedure

1. Open the **Forecasts** editor in one of the following ways:
 - With an active query in the Editor window, select **Query > Transfer To > Forecast**
 - Select **File > New > Other**

The New Wizard wizard opens. From the **QMF Objects** folder select the **Forecasts wizard**

- Click the **New Forecast** toolbar icon.

The Forecast editor opens in a separate tab.

2. Select the query or table that will be used to source the historical data.
3. Specify the **Date Parameters** options.
4. Specify the forecasting model's grouping.

The Grouping Hierarchy is used to specify what values are grouped and how they are ordered. For example, in wine sales, group by wine type and then location or by location then wine type.

5. Specify the query column that contains the values that will be forecast and the method of aggregation.
6. Specify the forecasting model's construction strategy, and distribution strategy:
 - The **Construction Strategy** is used to specify either a top-down or bottom-up approach, where the root node is at the top and the leaf node is at the bottom of a hierarchical tree diagram.
 - The **Distribution Strategy** specifies how the forecast values are distributed using the top-down construction strategy (e.g., from root to leaf nodes).
7. Specify **Forecasting Models** options including the forecasting models that are used, and their associated parameters.
8. Click **Run Forecast** on the toolbar.

The graphed forecast is displayed in the **Results** editor.

9. Experiment with one or more forecasting models to determine the best fit. Validation methods include:

- Validating the forecasting model using performance measures.

There are five performance measures that may be used as comparative criteria, including:

- Cumulative Forecast Error
- Mean Absolute Deviation
- Mean Square Error
- Mean Absolute Percent Error
- Tracking Signal

In most cases the Mean Square Error is used as the comparative criteria.

- Validating the forecasting model time series elements in accordance with any observed trend, seasonality, and cyclicity.

In the case of trend, an observed trend at a lower node may influence the choice of forecasting model for the entire forecast. For example, when marketing a new product it may be wiser to base the forecasting model on a smaller, more representative demographic with observable trends than use a larger, more diverse demographic with distorted or no observable trends.

10. Save the forecast to a file or to the repository.

Note: You can use saved forecasts as query objects when creating analytical queries, prompt hierarchies, drill down paths, quick reports, and visual projects.

Related tasks

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

About this task

To create a forecast and specify data source options:

Procedure

1. With an active data object that contains at least one date format column in the editor window, select **Query > Transfer To > Forecast**.
The **Forecast** editor opens. You can choose to embed a query or select a query to link to. An embedded data object is saved independently to the forecast. Any updates made to the original source after the forecast is created are not expressed. A linked data object creates a link between the data source and the saved forecast. Any updates made to the original source are automatically expressed in the forecast.
2. To specify a data object to embed, select **Embedded** and click **Import**.
The **Open** window opens.
3. Navigate to and select the data object that you want to embed in the forecast and click **Finish**.
The **Open** window closes.
4. Click **New**.
A new query opens in the **SQL** editor. Create the query that you want to embed in the forecast and save it.

5. Click **Edit**.

The embedded object opens in the editor. Edit the object and save it.

6. If you select **Linked**, click the ellipsis (...) next to the **Path** field.

The **Open** window opens.

7. Navigate to and select the object that you want to link to the forecast and click **Finish**.

The **Open** window closes and the path to the query is displayed in the **Path** field.

Note: Any query that you embed or link to must contain at least one date format column.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

About this task

To specify date parameters of the forecast:

Procedure

1. Open the **Group** page of the Forecast editor.
2. Select the date format column that you want to use to create the forecasted values from the **Date column** list.

Note: Only date format columns are available in the **Date column** list.

The following formats are available for date:

- **YYYYxMMxDD**
- **MMxDDxYYYY**
- **DDxMMxYYYY**
- **YYxMMxDD**
- **MMxDDxYY**
- **DDxMMxYY**

For time:

- **HHxMMxSS**
- **HHxMM**

For timestamp:

- **YYYYxMMxDD HHxMMxSS**
- **DDxMMxYYYY HHxMMxSS**
- **YYYYxMMxDD HHxMM**
- **DDxMMxYYYY HHxMM**
- **YYxMMxDD HHxMMxSS**
- **DDxMMxYY HHxMMxSS**
- **YYxMMxDD HHxMM**
- **DDxMMxYY HHxMM**

3. Select the historical data values that you want to use to calculate the forecasted values from the **Historical data** list. The available options are:
 - a) **All data:** Select this option to specify that all data values will be used to calculate the forecasted values.
 - b) **From:** Select this option to specify the starting data values that will be used to calculate the forecasted values. Enter the start date in the **From** calendar field. You can also enter the end date in the **To** calendar field. In this case, the data values retrieved between the start and end dates will be used to calculate the forecasted values.
 - c) **Last:** Select this option to specify that the data values of the last number of periods will be used to calculate the forecasted values. Specify the number of last periods in the **Periods** field.
4. Select the time period over which to aggregate data values from the **Period** list.
5. Specify the number of periods to calculate forecast values in the **Number of future periods** field.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

About this task

To specify grouping hierarchy options:

Procedure

1. Open the **Group** page of the Forecast editor
2. Select one or more columns from the **Available columns** list box and click the > button.
The column or columns are added to the **Selected columns** list box.

3. Select one or more columns from the **Selected columns** list box and click the < button. The column or columns are removed from the **Selected columns** list box and returned to the **Available columns** list box.
4. Click the >> or << buttons to add or remove all of the selected columns.
5. To change the order of the selected columns, click **Move Selected Column Up** or **Move Selected Column Down**.
6. Click the **Run** toolbar button to run the forecast.
The high level summary is displayed on the forecast.

Note: To see the results of other hierarchy levels, select the level that you want to display on the forecast in the **Outline** view.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

About this task

To specify strategy options:

Procedure

1. Open the **Model** page of the Forecast editor
2. Select the column that contains the values to forecast from the **Forecast column** drop-down list.
All available numeric columns are displayed in the **Forecast column** drop-down list.
3. Specify how the forecasted values will be aggregated in the **Aggregation strategy** drop-down list.
The aggregation strategy is largely dependent on the type of the value column that you are forecasting. In most cases you would use the **Group sum** option. There are instances however, such as aggregating temperatures, where you would use the **Group average** option.
4. Specify how the forecast model is constructed in the **Construction strategy** drop-down list.
If you select **Leaf**, the model is calculated at leaf level and aggregated upwards using a bottom-up approach. This strategy is most effective when there is a full history at the lower reporting level (for example, a rich data set with no nulls at the bottom level). If you select **Root**, the model is calculated at root level and distributed to leaf nodes using a top-down approach. A root level strategy is more effective when there is a smaller data set.
5. If you select the **Root** construction strategy, select how to distribute the root level forecast value to leaf levels in the **Distribution strategy** drop-down list.
Available values include:
 - **Equal** - Equally distributes the root level forecasts to each leaf level.
 - **Historical total** - Distributes root value forecasts proportionally to sub-levels according to the total contribution of each group across the whole historical data set.
 - **Most recent period** - Distributes root value forecasts to the most recent forecasted period.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

About this task

To specify forecasting models options:

Procedure

1. Open the **Model** page of the Forecast editor.
2. Click **Add**.
The **Forecast model selection** page of the **Forecast** wizard opens.
3. Select a forecast model.
Available forecast models include:
 - Simple moving average
 - Weighted moving average
 - Single exponential smoothing
 - Double exponential smoothing
 - Holt-Winters method
 - Multiplicative decomposition
 - Curve fitting
 - Polynomial regression
 - Neural network
4. Click **Next**.
The parameters page of the specified forecast model opens.
5. Specify the parameters of the forecast model and click **Finish**.
The **Forecast** wizard closes and the forecast model is displayed in the **Forecasting Models** table.
6. To delete a forecast model from the **Forecasting Models table**, click **Remove Model**.

7. To edit a forecast model parameters, perform the following steps:
 - a) Select a forecast model and click **Edit Model**. The parameters page of the selected forecast model opens.
 - b) Edit the parameters of the forecast model and click **Finish**. The **Forecast** wizard closes and the changes are saved.
8. To copy a forecast model, click **Copy Model**.

A duplicate model is created and displayed in the **Forecasting Model** table. This feature is useful if you want to test several models of the same type with slightly different parameters each time.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Setting simple moving average parameters

A moving average smooths a time series by forecasting the value for each period using the average of several previous periods.

About this task

To specify simple moving average parameters:

Procedure

1. Open the **Model** page of the Forecast editor.
2. Click **Add**.
The **Forecast model selection** page of the **Add Model** wizard opens.
3. Click **Simple moving average** and then click **Next**.
The **Simple moving average** page of the **Add Model** wizard opens.
4. Select the number of previous time periods to use to calculate the moving average from the **Number of periods** drop-down list.
5. Click **Finish**.
The **Add Model** wizard closes and the forecast model is displayed in the table.

Setting weighted moving average parameters

A weighted moving average smooths a time series by forecasting the value for each period using the weighted average of several previous periods.

About this task

To specify weighted moving average parameters:

Procedure

1. Open the **Model** page of the Forecast editor.
2. Click **Add**.
The **Forecast model selection** page of the **Add Model** wizard opens.
3. Click **Weighted moving average** and then click **Next**.
The **Weighted moving average** page of the **Add Model** wizard opens.
4. Specify the number of previous time periods to use to calculate the moving average in the **Number of periods** drop-down list.
5. Specify a weight for each of the previous periods in the **Weight** column of the table.
The periods are listed from the earliest to the most recent. For example, giving a high weight to the last period in the list will increase the importance of the most recent data relative to earlier data.
6. Click **Finish**.
The **Add Model** wizard closes and the forecast model is displayed in the table.

Setting single exponential smoothing parameters

A single exponential smoothing is a type of weighted moving average where the weights decrease exponentially for each successive time period.

About this task

To specify single exponential smoothing parameters:

Procedure

1. Open the **Model** page of the Forecast editor.
2. Click **Add**.

The **Forecast model selection** page of the **Add Model** wizard opens.

3. Click **Single exponential smoothing** and then click **Next**.

The **Single exponential smoothing** page of the **Add Model** wizard opens.

4. Specify a value in the **Exponential smoothing constant** field.

The number must be between zero and one. High values give recent periods more weight and low values give more weight to past data.

5. Click **Finish**.

The **Add Model** wizard closes and the forecast model is displayed in the table.

Setting double exponential smoothing parameters

A double exponential smoothing allows for exponentially decreasing weights that are applied to both previous values and previous trends between values.

About this task

To specify double exponential smoothing parameters:

Procedure

1. Open the **Model** page of the Forecast editor.
2. Click **Add**.

The **Forecast model selection** page of the **Add Model** wizard opens.

3. Click **Double exponential smoothing** and then click **Next**.

The **Double exponential smoothing** page of the **Add Model** wizard opens.

4. Specify a value in the **Exponential smoothing constant** field.

The number must be between zero and one. High values give recent periods more weight and low values give more weight to past data.

5. Specify a value in the **Trend smoothing constant** field.

The number must be between zero and one. High values give recent trends more weight and low values give more weight to past trends.

6. Click **Finish**.

The **Add Model** wizard closes and the forecast model is displayed in the table.

Setting Holt-Winters method parameters

The Holt-Winters method allows for a weighted exponential smoothing that takes seasonality into account.

About this task

To specify Holt-Winters method parameters:

Procedure

1. Open a forecast in the **Model** editor.
2. Click **Add**.

The **Forecast model selection** page of the **Add Model** wizard opens.

3. Click **Holt-Winters method** and then click **Next**.

The **Holt-Winters method** page of the **Add Model** wizard opens.

4. Select the frequency at which you expect to see seasonal effects on the data from the **Seasonal frequency** drop-down list.
5. Specify a value in the **Smoothing constant (alpha)** field.
The number must be between zero and one. High values give recent periods more weight and low values give more weight to past data.
6. Specify a value in the **Seasonal constant (beta)** field.
The number must be between zero and one. High values give recent periods more weight and low values give more weight to past data.
7. Specify a value in the **Trend constant (gamma)** field.
The number must be between zero and one.
8. Click **Finish**.
The **Add Model** wizard closes and the forecast model is displayed in the table.

Setting multiplicative decomposition parameters

A multiplicative decomposition extracts trend, seasonality, and cyclicity information from the data and uses these elements to predict future values.

About this task

To specify multiplicative decomposition parameters:

Procedure

1. Open a forecast in the **Model** editor.
2. Click **Add**.
The **Forecast model selection** page of the **Add Model** wizard opens.
3. Click **Multiplicative decomposition** and then click **Next**.
The **Multiplicative decomposition** page of the **Add Model** wizard opens.
4. Specify the frequency at which you expect to see seasonal effects on the data in the **Seasonal frequency** drop-down list.
5. Specify the frequency at which you expect to see cyclical effects on the data in the **Cycle length** drop-down list.
6. Click **Finish**.
The **Add Model** wizard closes and the forecast model is displayed in the table.

Setting curve fitting parameters

A curve fitting finds a mathematical function that most closely fits the data. This can be a line of best fit, exponential curve, or logarithmic curve.

About this task

To specify curve fitting parameters:

Procedure

1. Open a forecast in the **Model** editor.
2. Click **Add**.
The **Forecast model selection** page of the **Add Model** wizard opens.
3. Click **Curve fitting** and then click **Next**.
The **Curve fitting** page of the **Add Model** wizard opens.
4. Select a curve type radio button.
There are three types of curve types:

- **linear regression** - Select this option if the rate of change in forecast column values across periods is relatively constant.
 - **power regression** - Select this option if the rate of change in forecast column values across periods is increasing over time.
 - **logarithmic regression** - Select this option if the rate of change in forecast column values across periods is decreasing over time.
5. If you click **power regression**, select the exponential function that you want to use.
 6. Click **Finish**.

The **Add Model** wizard closes and the forecast model is displayed in the table.

Setting polynomial regression parameters

A polynomial regression finds a mathematical function of a certain order that most closely fits the data.

About this task

To specify polynomial regression parameters:

Procedure

1. Open the **Model** page of the Forecast editor.
2. Click **Add**.
The **Forecast model selection** page of the **Add Model** wizard opens.
3. Click **Polynomial regression** and then click **Next**.
The **Polynomial regression** page of the **Add Model** wizard opens.
4. Specify the order of the polynomial in the **Order** drop-down list.
5. Specify how the coefficients are set by clicking one of the radio buttons:
 - a) Click the first radio button to let the system calculate the coefficients.
 - b) Click the second radio button to manually set the coefficients.
6. If you click the second radio button, specify the value of each of the coefficients in the **Value** column of the table
7. Click **Finish**.

The **Add Model** wizard closes and the forecast model is displayed in the table.

Setting neural network parameters

A neural network uses a portion of the historic data to derive patterns that are then used to predict additional data.

About this task

The stage needs to be set just so.

Procedure

1. Open the **Model** page of the Forecast editor.
2. Click **Add**.
The **Forecast model selection** page of the **Add Model** wizard opens.
3. Click **Neural network** and then click **Next**.
The **Neural network** page of the **Add Model** wizard opens.
4. Specify a value in the **Allowable reduction (%)** field.
5. Specify a value in the **Allowable increase (%)** field.

6. Select the frequency at which you expect to see seasonal effects on the data from the **Seasonal frequency** drop-down list.
7. Specify how many times the neural network runs the data to train itself in the **Training iteration** drop-down list.
8. Specify the type of neural network algorithm to be used.
Available algorithms include:
 - **Gradient descent** - A first-order optimization algorithm.
 - **Quick prop** - A slight variation of the standard backpropagation of error algorithm.
 - **R-prop** - A resilient backpropagation of error algorithm.
9. Select the number of hidden neurons from the **Hidden neurons** drop-down list.
10. Specify the random seed of the neural network in the **Random seed** field.
11. Specify a value in the **Training set (%)** field.
12. Select the **Show training progress** check box to display the training progress of the neural network.
13. Click **Finish**.
The **Add Model** wizard closes and the forecast model is displayed in the table.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

About this task

To generate a summary of a forecast model's performance metrics:

Procedure

1. Open the forecast that you want to test.
2. Open a forecast in the **Forecast** editor.
3. Run the forecast model that you want to test.
The forecast results are displayed in the **Results** editor.

4. Click the **Create Performance**  toolbar button.

A table containing the forecast model's performance metrics is displayed.

Cumulative Forecast Error

Equal to the sum of differences between predicted and actual values.

Mean Absolute Deviation

Equal to the sum of the absolute values of the forecast error divided by the number of values. This metric tends to provide the best indicator of performance and is used as the default comparison criterion in forecast graphs.

Mean Square Error

Calculated as the sum (or average) of the squared error values. This performance metric is very sensitive to unique or large values, hence the error is amplified.

Mean Absolute Percent Error

Calculated as a percentage of the absolute difference between predicted and actual values divided by the number of values.

Tracking Signal

Calculated as a ratio of cumulative forecast error to the mean absolute deviation.

In general, the closer the error is to zero, the better the performance of the model (for example, a performance error equal to zero implies a perfect fit between the predicted and actual values).

- Optional: If you want to view the forecast hierarchical structure, open the **Outline** view by selecting **Window > Show view > Outline** from the main menu.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Observing trends

You use trends to track in what direction the forecasted data moves.

About this task

To observe the trend of the historical data:

Procedure

- Open the forecast for which you want to create a trend.
- Open a forecast in the **Forecast** editor.

3. Run the forecast model that you want to test.

The forecast results are displayed in the **Results** editor.

4. Click the **Create Trend**  toolbar button.

A trend graph and a trend table are displayed in the **Results** editor.

5. Observe the three types of trends in the trend chart to determine if there is a significant trend to the data, and in what direction it moves.

Local Trend

The difference between the current value within a period and the value for the last period.

Global Trend

The difference between the current value within a period and the average of all values. The period, order, and magnitude of fluctuation between positive and negative values can be used to determine the significance of the linear trend.

Linear Trend

The linear regression value for the current period, as displayed in the trend graph.

6. Optional: If you want to view the forecast hierarchical structure, open the **Outline** view by selecting **Window > Show view > Outline** from the main menu.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

[Transforming the Results tab into the Results editing view](#)

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

About this task

Some examples of seasonality might be a sharp escalation in retail sales in response to holiday shopping, increase in water consumption in the summer due to warmer weather, and increase in government spending at the end of the fiscal year. To observe seasonality within the historical data:

Procedure

1. Open the forecast for which you want to observe seasonality.
2. Open a forecast in the **Forecast** editor.
3. Run the forecast model that you want to test.

The forecast results are displayed in the **Results** editor.

4. Click the **Create Seasonality** toolbar button .

The **Create Seasonality** window opens.

5. Select the seasonality time period from the **Time period** drop-down list.
6. Select the historical data averaging method from the **Seasonality type** drop-down list.
7. Click **OK**.

The **Create Seasonality** window closes and a seasonality chart and seasonality graph are displayed in the **Results** editor. The seasonality table displays, for each period, the overall average for all seasons and the average for each seasonality period.

8. Optional: If you want to view the forecast hierarchical structure, open the **Outline** view by selecting **Window > Show view > Outline** from the main menu.

Related tasks

[Creating forecasts](#)

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

[Specifying forecast data source options](#)

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

[Specifying forecast date parameters](#)

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

[Specifying forecast grouping hierarchy options](#)

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

[Specifying forecast strategy options](#)

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

[Specifying forecasting models options](#)

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

[Monitoring forecast model performance](#)

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

About this task

Some examples of industries that experience cyclical demand are automobile, defense, mineral resources, and construction industries. To observe cyclicity within historical data:

Procedure

1. Open the forecast for which you want to observe cyclicity.
2. Open a forecast in the **Forecast** editor.
3. Run the forecast model that you want to test.

The forecast results are displayed in the **Results** editor.

4. Click the **Create Cyclicity**  toolbar button.

The **Create Cyclicity** window opens.

5. Select the cycle length from the **Number of points** drop-down list.

The cycle length is the number of periods expected in a complete cycle.

6. Click **OK**.

The **Create Cyclicity** window closes and a cyclicity chart and cyclicity table are displayed in the **Results** editor. The cyclicity table displays, for each period in a cycle, the overall average for all cycles and the average for each individual cycle. For example, if a cycle has twelve periods and the data history is three years then average values for three cycles are displayed.

7. Optional: If you want to view the forecast hierarchical structure, open the **Outline** view by selecting **Window > Show view > Outline** from the main menu.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

About this task

Once you run a forecast model, you can customize the generated chart formatting to your specifications. To edit a forecast result chart:

Procedure

1. Open a forecast in the **Forecast** editor.
2. Run the forecast model that you want to test.
The forecast results are displayed in the **Results** editor.
3. Click the **Chart Properties** icon in the menu bar.
The **Chart Properties** wizard opens.

4. Edit the chart formatting options and click **OK**.

The **Chart Properties** wizard closes and the changes are saved.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

About this task

Once you run a forecast model, you can customize the generated grid formatting to your specifications. To edit a forecast result grid:

Procedure

1. Open a forecast in the **Forecast** editor.

2. Run the forecast model that you want to test.
The forecast results are displayed in the **Results** editor.
3. Click the **Grid Properties** toolbar button.
The **Grid Properties** wizard opens.
4. Select **Columns** in the tree.
The **Columns** options are displayed.
5. Specify the columns to display in the grid by selecting the check boxes in the **Visible** column.
6. Select a **Value Column** in the tree.
The **Value Column** options are displayed.
7. Specify the column header name in the **Text** field.
8. Specify the font of the header name in the **Font** field.
9. Specify the background color of the column header cell in the **Background** field.
10. Specify the font of the text in the detail cells in the **Font** field.
11. Specify the format of the text in the detail cells in the **Format** field.
12. Specify the background color of the detail cells in the **Background** field.
13. Repeat the previous steps for each of the value columns.
14. Click **OK**.
The **Grid Properties** wizard closes and the changes are saved.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

[Specifying print options for a forecast](#)

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

[Transforming the Results tab into the Results editing view](#)

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

About this task

- To export a chart see the [“Exporting charts” on page 327](#) topic.
- To export query results to a database see the [“Exporting query results to a database” on page 198](#) topic.
- To export query results to a file see the [“Exporting query results to a file” on page 200](#) topic.

Related tasks

[Creating forecasts](#)

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

[Specifying forecast data source options](#)

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

[Specifying forecast date parameters](#)

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

[Specifying forecast grouping hierarchy options](#)

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

[Specifying forecast strategy options](#)

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

[Specifying forecasting models options](#)

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

[Monitoring forecast model performance](#)

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

[Observing trends](#)

You use trends to track in what direction the forecasted data moves.

[Observing seasonality](#)

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

[Observing cyclicity](#)

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

[Editing chart properties](#)

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

[Editing grid properties](#)

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

About this task

To control the appearance of printed pages for a forecast:

Procedure

1. With an active object in the editor window, open the **Page Setup** window by selecting **File > Page Setup**.
2. Select the printer that you want to use from the **Printer** list.
3. For detailed settings click **Properties**.
The **Page Setup** window opens.
4. Select the size of the printable area from the **Size** list.
5. Select the paper tray to use from the **Source** list.
6. Select the orientation of the printout from the **Orientation** area.
7. Specify the size of the margins of the page in the **Margins** fields.
8. Click **OK** to close **Page Setup** window.
9. In the **Page** area, specify the forecast objects that you want to print by selecting or clearing the **Chart** and **Table** check boxes.
10. In the same area, select one of the available page options:
 - a) **Print on separate pages** to print the forecast objects, the chart and the table, on separate pages.
 - b) **Print on the same page** to print both forecast objects on one page.
11. In the **Scale Options** area, select one from the available printing options:
 - a) **Fit to printable area** to fit the forecast size to the printable area.
 - b) **Scale print to %** to specify the required forecast size in percent.
12. To make the specified values default, click **Set as Default**.
13. Click **OK** to close **Page Setup** window.

The options that you specified are used to control the appearance of the printed object.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Transforming the Results tab into the Results editing view

You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

Transforming the Results tab into the Results editing view


You can detach the **Results** tab to view the processed results dynamically when you edit the QMF object.

About this task

You can detach the **Results** tab for the following objects:

- Analytical Query
- DBF, CSV, IXF, or XML file
- Dynamart
- Forecast
- Visual Query

To transform the **Results** tab into the view in three ways:

1. From the main menu, select **Window > Show View > Other > General > Results**.
2. From the main menu, select **Query > Detach Results View** or **Forecast > Detach Results View**.
3. Click the **Detach Results View** () toolbar button.

The **Results** view becomes available in a separate view.

To transform the **Results** view into the tab:

1. Close the **Results** view.
2. From the main menu, select **Query > Detach Results View** or **Forecast > Detach Results View**.

3. Click the **Detach Results View** () toolbar button.

The **Results** view transforms into the tab.

Related tasks

Creating forecasts

You can create a forecast using the Forecasts editor. You use the **Group** and **Model** pages of the **Forecasts** editor to set up a forecasting model.

Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

Specifying forecast grouping hierarchy options

You use the **Grouping Hierarchy** pane of the **Group** page to specify the columns that group the forecast data.

Specifying forecast strategy options

Use the **Strategy** pane of the **Forecast** editor to specify how the forecasted data is formatted.

Specifying forecasting models options

Use the **Forecasting Models** pane of the **Forecasts** editor to specify the forecasting models that are used to calculate the forecasted data.

Monitoring forecast model performance

You use forecast performance metrics to determine which model produces the best fit for the forecasted data.

Observing trends

You use trends to track in what direction the forecasted data moves.

Observing seasonality

Seasonality can be observed in a graph as a repeated pattern of regularly spaced peaks and troughs that have a consistent direction and approximately the same magnitude relative to the trend.

Observing cyclicity

Cyclicity allows you to observe periodic fluctuations that repeat within a value's time series, usually over a time period that is greater than one year.

Editing chart properties

You use the **Chart Properties** wizard to edit the properties of forecast result charts.

Editing grid properties

You use the **Grid Properties** wizard to edit the properties of forecast result grids.

Exporting forecasts

You can export charts and query result sets of generated forecasts to a file system or database using the **Export Chart** or **Export Query Results** wizard.

Specifying print options for a forecast

You can specify the page setup options that will be used when you print a forecast opened in the editor window.

Chapter 13. Working with charts

You can use charts for graphical representation of the data that is retrieved from queries.

About this task

You can create several different charts for a single result set of a visual query.

The following topics describe how to create and customize charts:

Creating charts

You can create a chart from a query result set with the help of the **New Chart** wizard.


Before you begin

You must run a query and have a query result set open on the **Results** tab.

About this task

To create a chart, do the following actions:

Procedure

1. Open the **New Chart** wizard in one of the following ways:
 - a) From the main menu, select **Results > Display Chart** and specify the type of the chart that you want to create. Then go to the step 3 of the procedure.
 - b) Click the **Display Chart** () toolbar button.
2. On the **Chart type** page, specify the type of the chart. For more information about chart types, see [“Specifying chart type options” on page 314](#) topic. Click **Next**.
3. On the **Data series** page, specify which columns of a result set are used to generate data points of a chart. For more information, see [“Specifying data series options” on page 314](#) topic. Click **Next**.
4. On the **Format** page, specify the options to define the visual output of the chart. For more information, see [“Specifying format options” on page 315](#) topic.
5. Click **Finish** to close the **New Chart** wizard and display the created chart.
6. To remove a chart that you created from a visual query results, see the [“Managing display modes” on page 324](#) topic.

Related tasks

[Creating charts from a query results grid](#)

You can create a ready-made chart by selecting the required columns from a query results grid. This allows you to review the returned data and decide what to display and how to display it.

[Working with charts in the Display editor](#)

You can create charts for a visual query result set and manage them using the **Display** editor.

[Editing charts](#)

You can edit charts and change their visual appearance by modifying the chart properties.

[Exporting charts](#)

You can export charts that are created from query result sets to a file system using the **"Export chart"** wizard.

[Specifying print options for charts](#)

For charts you can specify the page setup options for printing.

Specifying chart type options

You can specify how to display a chart on the **Chart type** page of the **New Chart** wizard. **Chart type** options determine the type of the chart and the visual format of the display mode.

About this task

To specify chart type options:

Procedure

1. From the **Chart type** list, select the required display mode for the chart. Available options are the following:
 - **Column** – Data points are represented by vertical columns.
 - **Bar** – Data points are represented by horizontal columns.
 - **Line** – Data points are plotted on the X-Y axes using interconnected lines.
 - **Area** – Data points are plotted similar to the line chart but with the area between the data series and x-axis filled with a specified color.
 - **Pie** – Data points are displayed as wedges of a pie.
 - **Scatter** – Data points are plotted as discrete points, using a user-configurable data symbol.
 - **Bubble** – Data points are plotted using spherical data symbols, each of which can be sized according to an expression or column value.
 - **Speedo** – Data points are displayed as points on a dial.

You can also select **Grid** mode to display data point in tabular grid.

2. From the **Dimension** list select the required dimension. Available variants are the following:
 - **2D** – The chart is displayed as a flat, two dimensional objects. It is available for all chart types.
 - **2D With Depth** – The chart is displayed as a two dimensional object, with data points displayed in three dimensions. It is available for Column, Bar, Area, and Pie chart types.

Note: Stacked Area Charts do not support **2D With Depth** dimension.
3. If you select Column, Bar, or Line chart type, select the required subtype. Available options are the following:
 - **Side-by-Side**
 - **Stacked**
 - **Percent Stacked**

Specifying data series options

You can specify which columns of a result set are used to generate data points of a chart on the **Data series** page of the **New Chart** wizard.

About this task

To specify the data series options of a chart:

Procedure

1. To specify the column that will populate the category axis, select a column from the **Result set** tree and click **Add to category**.
The column is added to the **Category Axis** node of the **Series** tree.

2. To specify a column that will be displayed as a value, select a column from the **Result set** tree and click **Add to value**.
The column is added to the **Values** node of the **Series** tree.
3. Select aggregation type for value column from the Aggregation drop-down list. Valid options are:
 - No aggregation
 - Count
 - First
 - Last
 - Maximum
 - Minimum
 - Sum
 - Average
 - Standard Deviation
4. Specify the way of aggregating a value column by selecting or clearing the **Calculate as formula** check box.
The **Calculate as formula** check box is available only if you use a calculated column as a value. The expression you have specified for the calculated column is used as aggregation formula.
If you select the **Calculate as formula** check box, the fields of the columns that provide data for calculated column (and thus for value column) are aggregated at first. The formula uses already aggregated field values as arguments to calculate the result value.
If you clear the **Calculate as formula** check box, the defined formula calculates the value column fields separately. After the calculations the fields are aggregated to calculate the result value.
5. To remove a column from the series tree, select the column and click **Remove**. The column is added back to the **Result set** tree.

Specifying format options

Specify **Chart Format** options to determine the visual formatting of the chart object.

About this task

On the **Format** page of the **New Chart** wizard you can specify Series options and Chart Area options.

Series options determine which category axis and value axis series are displayed in the chart, and how they are formatted.

Chart Area options determine the visual formatting of the chart object, such as the chart title, legend, and chart area color schemes.

Specifying Series chart format options

Series options determine which category axis and value axis series are displayed in the chart, and how they are formatted.

About this task

To specify or edit **Series** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, select **Series** in the tree.
The **Series** options are displayed.
2. Specify whether the chart data points will be colored according to the value series or the category in the **Color By** list.

3. To include or exclude one of the value series in the chart, select or clear the check box in the **Visible** column.

Specifying Category chart format options

Category chart format options control how the chart data points are displayed along the category axis.

About this task

To specify **Category** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, select **Category** in the tree.
The **Category** options are displayed.
2. Specify what data series is used to sort the data points along the category axis in the **Order By** list.
3. If you specified **Values** in the **Order By** list, select the value column that you want to use.
4. Specify whether the data points along the category axis are sorted ascending or descending in the **Order** list.
5. Select what type of cutoff that is used to restrict the display of data points along the category axis in the **Type** list.
Valid options include:
 - **No Cutoff** - No cutoff is used to restrict the category axis data points.
 - **Item Number** - Only the specified number of data points are displayed.
 - **Data Value(Less)** - Only rows that contain values that are larger than the specified value are displayed.
 - **Data Value(More)** - Only rows that contain values that are smaller than the specified value are displayed.
 - **Percentage(Less)** - Only rows that contain values that are more than the specified percentage of the data range.
 - **Percentage(More)** - Only rows that contain values that are less than the specified percentage of the data range.
6. Specify the value column that you want to use to determine cutoff in the **Value Series** list.
7. Specify the value that you want to use to determine the cutoff in the **Value** field.
8. Select the **Visible** check box to make excluded data points visible.
9. Specify what label is displayed with the excluded data points in the **Label** field.

Specifying Value(Y)Series chart format options

Value(Y)Series chart format options control how the value series axis is displayed.

About this task

To specify **Value(Y)Series** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, select **Value(Y)Series** in the tree.
The **Value(Y)Series** options are displayed.
2. Specify the title of the value series axis in the **Title** field.
3. To make the title of the value series axis visible, select the **Visible** check box.
4. Specify the color of the value series data points in the **Color** field.
5. Expand the **Value(Y)Series** node and select **Labels**.

6. Specify the labels options of the values series axis.

Related tasks

[“Specifying Labels chart format options” on page 319](#)

Labels chart format options control whether the labels of an axis are displayed and how they are formatted.

Specifying Chart Area chart format options

Chart Area chart format options control how the area of the chart is displayed.

About this task

To specify **Chart Area** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, select **Chart Area** in the tree.
The **Chart Area** options are displayed.
2. Specify the background color of the chart area in the **Background** field.
3. To display an outline around the chart area, select the **Visible** check box.
4. If you are editing the chart properties for a pie chart, select the **Auto** check box in the **Coverage** area to automatically make space for multiple pie charts.
The **Coverage** property of a pie chart controls how much of the proportion of the chart area individual pie charts inhabit. The **Auto** check box ensures that none of the individual pie charts in the chart area overlap.
5. If you clear the **Auto** check box, specify what proportion of the chart area that the pie chart takes up in the **Value** field.
For example, if you specify a value of 90 in the **Value** field, the pie chart will inhabit ninety percent of the total height of the chart area.

Specifying Axes chart format options

Axes chart format options control what axes of the chart are included, and how they are displayed.

About this task

To specify **Axes** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, select **Axes** in the tree.
The **Axes** options are displayed.
2. To include an axis in the chart, select the check box in the **Visible** column.
3. Specify the title of an axis in the **Title** column.
4. Select whether an axis will be linear or logarithmic in the **Type** column.
5. Expand the **Axes** node and select **X-Axis**.
6. Specify the **X-Axis** chart format options.
7. Specify the **Y-Axis-1** chart format options.
8. Specify the **Y-Axis-2** chart format options.

Specifying X-Axis chart format options

X-Axis chart format options control how the x-axis of the chart is displayed.

About this task

To specify **X-Axis** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, expand the **Axes** node, and select **X-Axis** in the tree.
The **X-Axis** options are displayed.
2. To display the x-axis of the chart, select the **Visible** check box.
3. To display the title of the x-axis, select the **Visible** check box next to the **Title** field.
4. Specify the title of the x-axis in the **Title** field.
5. Specify the font of the title in the **Font** field.
6. Specify the color of the x-axis in the **Color** field.
7. Select the position of the x-axis from the **Origin** list.

Valid options are:

- **Max** - The x-axis is located above the maximum value of the y-axis.
 - **Min** - The x-axis is located below the minimum value of the y-axis.
 - **Value** - The x-axis is flush against the bottom of the client area of the chart.
8. To display the values of the x-axis according to the category axis, select the **Is category axis** check box.
When this check box is selected, data points are displayed evenly along the x-axis, with every major tick mark representing the next data point down the line. When this check box is cleared, data points are displayed unevenly along the x-axis, with major tick marks placed at regular intervals.

Specifying Y-Axes chart format options

Y-Axis chart format options control how the y-axis of the chart is displayed.

About this task

To specify **Y-Axis** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, expand the **Axes** node, and select **Y-Axis** in the tree.
The **Y-Axis** options are displayed.
2. To display the y-axis of the chart, select the **Visible** check box.
3. To display the title of the y-axis, select the **Visible** check box next to the **Title** field.
4. Specify the title of the y-axis in the **Title** field.
5. Specify the font of the title in the **Font** field.
6. Specify the color of the y-axis in the **Color** field.
7. Specify the position of the y-axis in the **Origin** list.
Valid options are:
 - **Max** - The y-axis is located after the maximum value of the x-axis.
 - **Min** - The y-axis is located before the minimum value of the x-axis.
 - **Value** - The y-axis is flush against the side of the client area of the chart.
8. Specify whether the y-axis values will be displayed as a linear function or a logarithmic function in the **Type** list.

Specifying Gridlines chart format options

Gridlines chart options control whether gridlines are displayed and how they are formatted.

About this task

To specify **Gridlines** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, expand the **Axes** node.
2. Expand the **X-Axis** or **Y-Axis** node and select **Gridlines** in the tree.
The **Gridlines** options are displayed.
3. To display the major gridlines, select the **Visible** check box in the **Major Grid** section.
4. Specify the color of the major gridlines in the **Color** field.
5. To display the minor gridlines, select the **Visible** check box in the **Minor Grid** section.
6. Specify the color of the minor gridlines in the **Color** field.

Specifying Labels chart format options

Labels chart format options control whether the labels of an axis are displayed and how they are formatted.

About this task

To specify **Labels** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, expand the **Axes** node.
2. Expand the **X-Axis** or **Y-Axis** node and select **Labels** in the tree.
The **Labels** options are displayed.
3. To display the labels of an axis, select the **Visible** check box.
4. Specify the font of the axis labels in the **Font** field.
5. Click the ellipsis (...) next to the **Format** field.
The **Choose a format** window opens.
6. Select the format of the column cell values from the **Format** list.

The following formats are available for numeric data type:

- **As is** - The exact value of the column is displayed without any formatting.
- **Currency** - Column values are displayed as currency. The locale settings of the current session determine the currency unit that is displayed.
- **Decimal** - Column values are displayed as decimals.
- **Scientific** - Column values are displayed in scientific notation.
- **Percent** - Column values are displayed as percentages.
- **Custom** - Data point values are displayed according to the pattern that you specified in the **Pattern** field.

The following formats are available for date data type:

- **As is** - The exact value of the column is displayed without any formatting.
- **Database** - The date is displayed in the format of the database.
- **Date** - Select the date presentation from the available formats.
- **Custom** - The date is displayed according to the pattern that you specified in the **Pattern** field.

The following formats are available for timestamp data type:

- **As is** - The exact value of the column is displayed without any formatting.
- **Database** - The timestamp is displayed in the format of the database.
- **Timestamp** - Select the timestamp presentation from the available formats.
- **Date** - Only the date component of the timestamp is displayed.
- **Time** - Only the time component of the timestamp is displayed.
- **Custom** - The timestamp is displayed according to the pattern that you specified in the **Pattern** field.

7. To include thousands separators with column values, select the **Thousands separator** check box.

8. Specify the number of decimal places that are displayed in the **Decimal places** field.

A value of '0' indicates that no decimal places are displayed.

9. Click **OK** to close the **Choose a format** dialog.

10. Specify the angle of the labels of the axis in the **Rotation** field.

A value of '0' indicates no rotation. Labels are displayed with no angle. Higher numbers indicate higher angles. Values can be both positive and negative. Positive numbers turn labels counter clockwise and negative values turn labels clockwise.

Specifying Range chart format options

Range options allow you to manually format the range and step count of an axis of a chart.

About this task

To specify **Range** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, expand the **Axes** node.
2. Expand the **X-Axis** or **Y-Axis** node and select **Range** in the tree.
The **Range** options are displayed.
3. To manually set the range of an axis, select the **Is manual Range** check box.
4. To specify the number of steps on the axis, click **Step count** and specify the number of steps in the field.
5. To specify the size of each step on the axis, click **Step size** and specify the size of each step in the text field.
6. Specify the minimum value of the range of the axis in the **Minimum** field.
7. Specify the maximum value of the range of the axis in the **Maximum** field.

Specifying Title chart format options

Title chart format options control whether the title of the chart is displayed and how it is formatted.

About this task

To specify **Title** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, select **Title** in the tree.
The **Title** options are displayed.
2. To display the title of the chart, select the **Visible** check box.

3. From the **Type** drop-down list, select **Automatically Generated** if you want the chart title to be generated automatically by using the axes columns. If you want to specify the chart title by hand, select **Custom Text**.

4. If you selected **Custom Text**, specify the title of the chart in the **Text** field.

You can add a prompt value to a chart title to update the title automatically when a different prompt value is specified.

For example: if your chart is based on the Q.STAFF table and you want to insert the PRMPT simple prompt value in the title of your chart, your expression might be: %G for DEPT № = &PRMPT, where DEPT № is the description text for the PRMPT simple prompt value and %G is the variable that puts the names of the category column and the value column of your chart into the chart title. Also, you can use the %D variable, that puts the current date in the title of your chart.

5. Specify the font of the title in the **Font** field.

6. Select the location of the chart from the **Location** list.

Specifying Legend chart format options

Legend chart format options control whether a legend is displayed and how it is formatted.

About this task

To specify **Legend** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, select **Legend** in the tree.

The **Legend** options are displayed.

2. To display the legend of the chart, select the **Visible** check box.

3. To display the title of the legend, select the **Visible** check box that is next to the **Title** field.

4. Specify the title of the legend in the **Title** field.

5. Specify the font of the title in the **Font** field.

6. Specify the position of the legend in the **Position** drop-down list.

7. Specify the font of the text of the legend in the **Font Text** field.

8. To display an outline around the legend, select the **Visible** check box that is next to the **Outline** field.

Specifying Plot Area chart format options

Plot Area chart format options control the look and feel of the area of the chart that contains the data axes and the client area.

About this task

To specify **Plot Area** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, select **Plot Area** in the tree.

The **Plot Area** options are displayed.

2. Specify the background color of the plot area in the **Background** field.

3. To display an outline around the plot area, select the **Visible** check box that is next to the **Outline** field.

4. Specify the distance that you would like to inset the chart area from the edge of the plot area in the **Insets(points)** fields.

Specifying Client Area chart format options

Client Area chart format options control how the client area of the chart is displayed.

About this task

To specify **Client Area** chart format options:

Procedure

1. On the **Format** page of the **New Chart** wizard, select **Client Area** in the tree.
The **Client Area** options are displayed.
2. Specify the background color of the client area in the **Background** field.
3. To display an outline around the client area, select the **Visible** check box that is next to the **Outline** field.

Creating charts from a query results grid

You can create a ready-made chart by selecting the required columns from a query results grid. This allows you to review the returned data and decide what to display and how to display it.

Before you begin

You must run a visual query and have a query results grid open on the **Results** tab.

About this task

To create a chart from a query results grid:

Procedure

1. Select the columns that you want to include in the chart by holding the **Ctrl** key and clicking each column.
2. Click the down arrow next to the **Display Chart** toolbar button and select the type of chart that you want to display.
The **Data series** page of the **New Chart** wizard opens. The columns that you selected are displayed in the **Series** tree.
Note: Columns are automatically organized into the **Category Axis** and **Values** nodes depending on their data types. Category columns are the columns that contain character data, and value columns are the columns that contain numerical data. For example, if you selected a column of department names and a column of yearly salary totals, the department name column would automatically be placed under **Category Axis**, and the salary column would automatically be placed under **Values**.
3. Select each value column and select how it is aggregated from the **Aggregation** list.
4. Click **Next**.
The **Format** page opens.
5. Specify the different formatting options of the chart and click **Finish** to close the **New Chart** wizard and display the created chart. The chart is displayed on the **Results** tab.

Related tasks

[Creating charts](#)

You can create a chart from a query result set with the help of the **New Chart** wizard.

[Working with charts in the Display editor](#)

You can create charts for a visual query result set and manage them using the **Display** editor.

[Editing charts](#)

You can edit charts and change their visual appearance by modifying the chart properties.

[Exporting charts](#)

You can export charts that are created from query result sets to a file system using the **"Export chart"** wizard.

[Specifying print options for charts](#)

For charts you can specify the page setup options for printing.

Working with charts in the Display editor

You can create charts for a visual query result set and manage them using the **Display** editor.

About this task

All charts that you create for a visual query result set are stored with this result set. You use the **Display** editor to manage the display modes of a query result set. The **Display** editor contains the following panes:

- Result set
- Layout structure
- Display modes

The **Result set** pane displays all available result set columns regardless of the selected display mode. Query result columns are automatically grouped into three types: **Category columns**, **Value columns**, and **Columns with aggregates**. Category columns are columns that contain character data, and it is suggested that they are used for non-numerical data axes, such as the x-axis of a column chart. Value columns are columns that contain numerical data, and it is suggested that they are used for numerical data axes, such as the pie wedges of a pie chart. Columns with aggregates are copies of any columns to which the aggregations are applied. You can use the **Result set** pane to add groupings and aggregations to any column in the result set.

The **Layout structure** pane displays the specific layout structure of the selected display mode. Unlike the **Result set** pane, the **Layout structure** pane displays only the columns, groupings, and aggregations of the selected display mode. You can use the **Layout structure** pane to edit the individual layout structures the display modes.

The **Display modes** pane displays a list of available display modes. Each display mode has an independent layout structure and display settings. You can use the **Display modes** pane to add, delete, and edit the display modes of the given query result set.

Related tasks

[Creating charts](#)

You can create a chart from a query result set with the help of the **New Chart** wizard.

[Creating charts from a query results grid](#)

You can create a ready-made chart by selecting the required columns from a query results grid. This allows you to review the returned data and decide what to display and how to display it.

[Editing charts](#)

You can edit charts and change their visual appearance by modifying the chart properties.

[Exporting charts](#)

You can export charts that are created from query result sets to a file system using the **"Export chart"** wizard.

[Specifying print options for charts](#)

For charts you can specify the page setup options for printing.

Creating charts in the Display editor

You can create a ready-made chart for a visual query result set by selecting the columns in the **Display** editor.


Before you begin

You must run a visual query and switch to the **Display** tab.

About this task

To create a chart in the **Display** editor, do the following actions:

Procedure

1. In the **Result set** pane select the columns that you want to include in the chart by holding the **Ctrl** key and clicking each column.
2. With the mouse button pressed, drag the columns to the **Chart** tree in the **Display modes** pane. Release the mouse button.
A new chart appears in the **Chart** tree. This chart becomes default.
3. You can modify the created chart. For more information, see the [Editing charts](#) topic.
4. Click the **Run Query**  toolbar button to display the created chart on the **Results** tab.

Results

The chart that you created for a visual query result set is stored with this result set.

Managing display modes

You use the **Display modes** pane of the **Display** editor to add, edit, and delete the display modes of a query result set.

About this task

With the display modes you can visualize result set data in many different ways. You can create any number of display modes with different content and visual formatting. Every result set starts with a grid called **Results grid**. This is the standard query results grid that is displayed when you run a query for the first time. You can not edit this grid, but you can add a new one to the **Grid** node of the **Display modes** tree, and edit its layout as needed.

To manage the display modes of a query result set:

Procedure

1. Navigate to the query that has the result set that you want to edit and open it in the **Visual Query** editor.
2. Click the **Display** tab.
The **Display** editor opens.
3. To add a display mode, right-click either **Grid** or **Chart**, select **New**, and select the display mode that you want to add.
4. To rename a display mode, right-click it and select **Rename** from the pop-up menu.
5. To edit a display mode, right-click it and select **Edit** from the pop-up menu.
6. To delete a display mode, right-click it and select **Remove** from the pop-up menu.
7. To move a display mode up or down in the list, right-click it and select **Move Up** or **Move Down** from the pop-up menu.
8. To set a given display mode as default, right-click it and select **Make Default** from the pop-up menu.
The default display mode will open first every time the query is run.

Editing query result set columns

You use the **Result set** pane of the **Display** editor to modify the columns of a result set.

About this task

To modify result set columns:

Procedure

1. Navigate to the query that has the result set that you want to edit and open it in the **Visual Query** editor.
2. Click the **Display** tab.
The **Display** editor opens.
3. To roll up a column into the top group, right-click the column and select **Top Group** from the menu.
4. To roll up a column into the side group, right-click the column and select **Side Group** from the menu.
5. To add aggregation to a column, right-click the column and select an aggregation type from the menu.
Available aggregations are:
 - Average
 - Count
 - Cumulative Percentage of Group
 - Cumulative Percentage of Total
 - Cumulative Sum
 - First
 - Last
 - Maximum
 - Minimum
 - Percentage of Group
 - Percentage of Total
 - Standard Deviation
 - Sum
6. To remove aggregation from a column, right-click the column and select **No Aggregation** from the menu.
7. To add a count column to the layout structure of the selected display mode, right-click the **(count)** column in the **Columns with aggregates** node and select **To Layout structure** from the menu.

Editing display mode layout structures

You use the **Layout structure** pane of the **Display** editor to edit the layout structure of individual display modes.

About this task

To modify the layout structure of your query result set:

Procedure

1. Navigate to the query that has the result set that you want to edit and open it in the **Visual Query** editor.
2. Click the **Display** tab.
The **Display** editor opens.
3. Select the display mode that you want to edit.
The layout structure of the selected display mode opens in the **Layout structure** pane.
4. To roll up a column into the top group or side group, click and drag the column to the **Top Groups** or **Side Groups** nodes.
5. To add aggregation to a column, right-click the column and select an aggregation type from the menu.
Available aggregations are:
 - Average

- Count
 - Cumulative Percentage of Group
 - Cumulative Percentage of Total
 - Cumulative Sum
 - First
 - Last
 - Maximum
 - Minimum
 - Percentage of Group
 - Percentage of Total
 - Standard Deviation
 - Sum
6. To remove aggregation from a column, right-click the column and select **No Aggregation** from the pop-up menu.
 7. To remove a column from the layout structure, right-click the column and select **Remove** from the pop-up menu.
 8. To change the order of the columns, click a drag the columns to place them in the order that you want them to appear in the result set.
 9. If you are editing the layout structure of a chart, you can click and drag columns to move them to and from the **Category Axis** and **Values** nodes.

Editing charts

You can edit charts and change their visual appearance by modifying the chart properties.

About this task

To edit a chart, perform the following procedure:

Procedure

1. Open the **Edit chart properties** wizard in one of the following ways:
 - Open a chart on the **Results** tab and select **Results > Chart properties** from the main menu.
 - In the **Display modes** pane, right-click a chart that you want to edit and select **Edit** from the menu.
2. In the **"Edit chart properties"** wizard, switch to the appropriate tab to edit chart type, data series, or format options.

For more information about chart properties, see the following topics:

- [Specifying chart type options](#)
- [Specifying data series options](#)
- [Specifying format options](#)

3. Click **OK** to save changes and close the **"Edit chart properties"** wizard.

Related tasks

[Creating charts](#)

You can create a chart from a query result set with the help of the **New Chart** wizard.

[Creating charts from a query results grid](#)

You can create a ready-made chart by selecting the required columns from a query results grid. This allows you to review the returned data and decide what to display and how to display it.

[Working with charts in the Display editor](#)

You can create charts for a visual query result set and manage them using the **Display** editor.

Exporting charts

You can export charts that are created from query result sets to a file system using the **"Export chart"** wizard.

Specifying print options for charts

For charts you can specify the page setup options for printing.

Exporting charts

You can export charts that are created from query result sets to a file system using the **"Export chart"** wizard.

Before you begin

Open a chart that you want to export on the **Results** tab of a visual query results.

About this task

To export a chart:

Procedure

1. Open the **"Export chart"** wizard in one of the following ways:
 - With a chart opened in the editor, select **File > Export** to open the **Select** page of the **Export** wizard, and then select **Chart** from the list of available export wizards.
 - With a chart opened in the editor, click the **Export** toolbar button.
2. Select the source object from the **Sources** list.
3. From the **Export Type** list select the type of the exported file. The following formats are available:
 - PDF
 - PNG
4. Click **Next**. The second page of the **"Export chart"** wizard opens.
5. In the **Select export parameters** section, specify the size of the image.
6. Click **Update Chart** to refresh the **Chart export preview** window.
7. Click **Finish** to

Related tasks

Creating charts

You can create a chart from a query result set with the help of the **New Chart** wizard.

Creating charts from a query results grid

You can create a ready-made chart by selecting the required columns from a query results grid. This allows you to review the returned data and decide what to display and how to display it.

Working with charts in the Display editor

You can create charts for a visual query result set and manage them using the **Display** editor.

Editing charts

You can edit charts and change their visual appearance by modifying the chart properties.

Specifying print options for charts

For charts you can specify the page setup options for printing.

Specifying print options for charts

For charts you can specify the page setup options for printing.

Before you begin

If a chart is created from visual query results, open it on the **Results** tab.

About this task

To control the appearance of printed pages for a chart:

Procedure

1. From the main menu, select **File > Page Setup** to open the **Page Setup** window.
2. Select the printer that you want to use from the **Printer** list.
3. For detailed settings click **Properties**.
4. Select the size of the printable area from the **Size** list.
5. Select the paper tray to use from the **Source** list.
6. Select the orientation of the printout from the **Orientation** area.
7. Specify the size of the margins of the page in the **Margins** fields and click **OK**.
8. In the **Scale Options** area, select one of the available printing options:
 - a) **Fit to printable area** to fit the chart size to the printable area.
 - b) **Scale print to %** to specify the required chart size in percent.
9. To make the specified values default, click **Set as Default**.
10. Click **OK** to close the **Page Setup** window.

The options that you specified are used to control the appearance of the printed object.

Related tasks

Creating charts

You can create a chart from a query result set with the help of the **New Chart** wizard.

Creating charts from a query results grid

You can create a ready-made chart by selecting the required columns from a query results grid. This allows you to review the returned data and decide what to display and how to display it.

Working with charts in the Display editor

You can create charts for a visual query result set and manage them using the **Display** editor.

Editing charts

You can edit charts and change their visual appearance by modifying the chart properties.

Exporting charts

You can export charts that are created from query result sets to a file system using the **"Export chart"** wizard.

Chapter 14. Working with drill-down paths

A drill-down path is an interactive series of charts that allows you to display several levels of details of a given query.

About drill-down paths

A drill-down path is an interactive series of charts that allows you to display several levels of details of a given query.

Drill-down paths allow you to drill down into several different views of query results. Each level of detail is represented by a different display mode, and each display mode chart can be fully customized. The data that is passed from one level of detail to another is driven by user interaction with the data points of the display mode charts. The movement of data from one level to another is called an analysis path.

For example, you create a column chart that displays the sum of the salaries of employees in each department of a company. The column chart displays a different column for each department. Next, you create a pie chart that displays the sum of the salaries of all of the employees of a specific department, divided up by job type. The pie chart displays a different pie wedge for each job type. The analysis path passes data from the specified department column to the next chart. Now, when you click a specific column of the first chart, the specified department is displayed in the pie chart. You can go back to the original column chart and click a different department's column. The pie chart now displays the new department's data points. You can create any number of analysis paths to provide more specific views of the query data.

Related tasks

[Creating drill-down paths](#)

Drill-down paths allow you to drill down into several different views of query results.

[Managing drill-down path recording display modes](#)

You can use the **Drill-down structure** pane of the drill-down path **Design** editor to manage the display modes.

[Specifying print options for a drill-down path](#)

You can specify the page setup options that will be used when you print a drill-down path opened in the editor window.

Creating drill-down paths

Drill-down paths allow you to drill down into several different views of query results.

About this task

To create a drill-down path, perform the following procedure:

Procedure

1. Open the **Create New Drill-down Path** wizard in one of the following ways:
 - With an active query in the editor window, select **Query > Transfer To > Drill-down Path**
 - Select **File > New > Other**
 - From the **QMF Objects** folder, select the Drill-down Path wizard
 - Click the **New Drill-down Path** toolbar button.

The **Create New Drill-down Path** wizard opens.

If you opened the Create New Drill-down Path wizard by selecting **Query > Transfer To > Drill-down Path**, the **Chart Type** page of the wizard opens.

Otherwise, the Create New Drill-down Path page opens and you must perform the following actions:

- Specify a name for the drill-down path in the **Name** field
 - In the **Data Source** section, specify a query to use.
 - To use a saved in a repository or file query click ellipsis (...) right to the **Saved query**.
 - To use an opened query select one of the opened queries in the corresponding list box.
 - Click **Next**. The Chart Type page of the wizard opens.
2. Specify chart type options and click **Next**.
 - The **Data Series** page opens.
 3. Specify data series options and click **Next**.
 - The **Format** page opens.
 4. Specify format options and click **Finish**.
 - The **Create New Drill-down Path** wizard closes and the first chart of the drill-down path is displayed in the **Results** editor.
 5. To add another level to the drill-down path, click one of the data points of the chart.
 - The **New drill-down path node** wizard opens.
 6. Specify chart type, data series, and chart format options and click **OK**.
 - The new chart is displayed in the **Results** editor. The chart that you create is automatically parameterized to accept data from the previous chart. For example, if the first chart displayed data values from a column of department names, the new chart will only display data values from a specific department.
 7. Repeat this process for every level of detail that you want to include in the drill-down path and click the **Stop recording drill-down path** toolbar button.
 - You are prompted to confirm that you want to stop recording the drill-down path.
 8. Click **No** to discard the recording.
 - All of the previous levels and charts are discarded.
 9. Click **Yes** to save the recording.
 - The drill-down path and all of its levels and charts are saved.
 10. Once the drill-down path is saved, navigate between charts using the **Switch to the Previous Scene** and **Switch to the Next Scene** toolbar buttons.
 11. To edit a specific chart, navigate to the chart and click the **Drill-down path node properties** toolbar button.
 - The **Drill-down Level Properties** window opens.
 - For more information about editing chart properties, see the following topics:
 - [Specifying chart type options](#)
 - [Specifying data series options](#)
 - [Specifying format options](#)
 12. Edit the chart properties that you want to change and click **OK** to save the changes and close the **Drill-down Level Properties** window.
 13. To record a new drill-down path using an existing chart as the base chart, navigate to the chart that you want to start with and click the **Record drill-down path** toolbar button.
 - The **New drill-down path node** wizard opens, allowing you to create a new chart level. The new drill-down path will start with the chart that was open when you clicked the toolbar button, and the new chart will be the next level down.
 14. To record a new branch of a drill-down path, navigate to the chart in the current drill-down path recording that you want to branch from and click the **Branch out drill-down path and switch to the record mode** toolbar button.
 - The **Edit drill-down path node** wizard opens, allowing you to change the next chart level down from where you clicked the toolbar button. This creates a new drill-down path recording that retains all

of the previous chart levels from the start to the point at which you clicked the toolbar button, but allows you to record new chart levels that branch out from the original recording. The original saved recording is preserved.

Related concepts

[About drill-down paths](#)

A drill-down path is an interactive series of charts that allows you to display several levels of details of a given query.

Related tasks

[Managing drill-down path recording display modes](#)

You can use the **Drill-down structure** pane of the drill-down path **Design** editor to manage the display modes.

[Specifying print options for a drill-down path](#)

You can specify the page setup options that will be used when you print a drill-down path opened in the editor window.

Managing drill-down path recording display modes

You can use the **Drill-down structure** pane of the drill-down path **Design** editor to manage the display modes.

About this task

Each level of detail of a drill-down path recording is represented by a different display mode. Each drill-down path display mode is edited separately, allowing you to display data using different charts and formatting options. You use the **Drill-down structure** pane of the drill-down path **Design** editor to manage the available display modes and the analysis paths that they represent.

To manage drill-down path display modes:

Procedure

1. Open a drill-down path in the drill-down path **Design** editor.
2. In the **Drill-down structure** pane, right-click a display mode, select **New**, and specify the type of display mode that you want to add.
3. From the **Data source** list select a query, dynamart, table or forecast, on the basis of which you want to create a drill-down level.
4. To add a new object click **Add data source object**. In the **Set Data Source** dialog select the necessary data source object type and specify a path to it in the **Path** field:
 - a) **Embedded** - the object is saved in the drill-down path, so it can be transferred with it to a file or repository when you save this drill-down.
 - b) **Linked** - the drill-down path saves a link to the data source object. The object is not saved to a file or repository with a drill-down path.
5. If you add a new data source object, in the **Relation between data sources** area select a column of the parent object from the **Master column** list and a column from the child object from **Detail column** list. These two columns are used to join the data from two objects.
6. If the specified columns have **TIMESTAMP** format you can select the necessary value from **Calendar function** list to specify the format settings of the columns. Calendar function joins columns with **TIMESTAMP** data from different data sources if they belong to one and the same time period, for example, one year, quarter or month.
7. Click **Next** to open **Chart Type page** of the **New Drill-down Path Display Mode** wizard.
8. Specify chart type options and click **Next**.
The **Data Series** page options.
9. Specify data series options and click **Next**.
The **Format** page opens.

10. Specify format options and click **Finish**.

The **Create New Drill-down Path** wizard closes and the first chart is displayed in the **Drill-down structure** pane of the **Drill-down Path** editor.

Note: The new display mode is placed underneath the right-clicked display mode, and represents the next level of detail in the drill-down structure.

11. Right-click a display mode and select **Make Default**.

The default display mode is the first chart that is displayed when you view query results

12. Right-click a display mode and select **Edit** from the pop-up menu.

The **Edit display mode** wizard opens.

13. Make the necessary changes to the chart type, data series, and chart format options and click **OK**.

The **Edit display mode** wizard closes.

14. Right-click a display mode and select **Remove** from the pop-up menu to delete it from the **Drill-down structure** tree.

15. Right-click a display mode and select **Move Up** or **Move Down** to change its location in the **Drill-down structure** tree.

Note: You can select the necessary level of detail to be displayed in the drill-down by clicking a proper chart area. The **Drill-down** window with a list of available sublevels appears. Select the necessary one and click **OK**. You cannot select a level if there is only one sublevel or the drill-down path is being recorded.

Related concepts

[About drill-down paths](#)

A drill-down path is an interactive series of charts that allows you to display several levels of details of a given query.

Related tasks

[Creating drill-down paths](#)

Drill-down paths allow you to drill down into several different views of query results.

[Specifying print options for a drill-down path](#)

You can specify the page setup options that will be used when you print a drill-down path opened in the editor window.

Specifying print options for a drill-down path

You can specify the page setup options that will be used when you print a drill-down path opened in the editor window.

About this task

To control the appearance of printed pages for a drill-down path:

Procedure

1. With an active object in the editor window, open the **Page Setup** window by selecting **File > Page Setup**.
2. Select the printer that you want to use from the **Printer** list.
3. For detailed settings click **Properties**.
The **Page Setup** window opens.
4. Select the size of the printable area from the **Size** list.
5. Select the paper tray to use from the **Source** list.
6. Select the orientation of the printout from the **Orientation** area.
7. Specify the size of the margins of the page in the **Margins** fields.
8. Click **OK** to close **Page Setup** window.

9. In the **Scale Options** area, select one from the available printing options:
 - a) **Fit to printable area** to fit the drill-down path size to the printable area.
 - b) **Scale print to %** to specify the required drill-down path size in percent.
10. To make the specified values default, click **Set as Default**.
11. Click **OK** to close **Page Setup** window.

The options that you specified are used to control the appearance of the printed object.

Related concepts

About drill-down paths

A drill-down path is an interactive series of charts that allows you to display several levels of details of a given query.

Related tasks

Creating drill-down paths

Drill-down paths allow you to drill down into several different views of query results.

Managing drill-down path recording display modes

You can use the **Drill-down structure** pane of the drill-down path **Design** editor to manage the display modes.

Chapter 15. Working with visual projects

Visual projects include visual reports and visual dashboards.

About this task

Visual reports

Visual reports are page-based, printable reports that include both formatted text and graphics to display persistent data to a wide variety of users. Visual reports can also contain data driven graphics (such as maps and charts) inserted in different sections (such as the headers or footers) of the report. Each of the data driven graphics can present data from multiple queries that are run across the enterprise.

Ad hoc reports

Ad hoc reports provide a flexible and user-friendly approach to creating reports, while making the overall design process more efficient. They include a scene with a canvas on which you create a report. A canvas can contain data objects such as charts, queries, and dynamarts.

Visual dashboards

Visual dashboards present interactive or persistent data obtained from querying multiple heterogeneous data sources across the enterprise. Visual dashboards present data in a scene format using a wide variety of graphics including charts, graphs, maps and user interface widgets. Data driven graphical objects can be easily linked so that user selections will trigger unique data displays. Content developers can create a visual dashboard that can be viewed by multiple users with either QMF for Workstation or QMF for WebSphere.

Visual applications

You cannot create visual applications in QMF for Workstation starting from Fix Pack 4. In current release, you can run visual applications that were created in previous versions of QMF. The existing visual applications are automatically converted to visual dashboards, if you open them for editing. After conversion the initial visual application remains unchanged. You can edit the created visual dashboard, and then save it to a repository or file. In this case, the created visual dashboard cannot be run in previous versions of QMF.

The following functionality specific to visual applications can be now used in visual dashboards:

- Canvas object
- Tree object
- Show dialog event action
- Close dialog event action

Visual project templates

Templates provide standardized, pre-formatted examples of visual projects (visual reports and dashboards). They are reusable components that facilitate the design process.

Visual project designers might use templates for a visual project as the entry-point into their project development process.

Templates help developers avoid repetitive development and promote effective reuse strategies. For example, if an enterprise has multiple visual dashboards that connect to a particular data source, rather than defining the connection properties in multiple visual dashboards, you could create a single template with the required connection. The new visual dashboards that you create can be based on this one template.

There are three template types for visual projects:

- Predefined

- Local
- Repository

Predefined templates

Predefined templates are included with QMF for Workstation.

The layout of predefined templates cannot be changed.

Currently there is a single predefined default template included with QMF for Workstation. The name of the default predefined template is **Blank**. The **Blank** template contains no structural elements.

Predefined templates provide a base-level functionality for a yet to be built business function. For example, you can use a predefined template as the basis for a connecting to a data source, running a standard set of queries or a customized set of charts and tables, connected with lists or other data driven controls.

Local templates

Local templates are those that have been created for your organization and saved *locally*.

Each template contains predefined content (such as company logos, standardized text, frequent graphical layouts).

The templates are listed by category for organizational purposes.

Typically, local templates are used by individual developers for their own purposes. Visual projects saved as local templates can not be shared via repository services.

Repository templates

Repository templates are those that have been created for your organization and saved in a *shared repository*.

Each template contains predefined content (such as company logos, standardized text, frequent graphical layouts).

The templates are listed by category for organizational purposes.

Users can edit the structural content of repository templates to suit their needs.

Typically, repository templates are used by multiple developers. Visual projects saved as repository templates are made available to multiple developers through repository services.

Creating template categories for visual projects

Use the **Create New [objectname] Template Category** wizard to create a template category for your visual reports and dashboards.

About this task

To create a template category for a visual project (visual report or dashboard), do the following steps

Procedure

1. From the main menu, select **File > New > Other > QMF Objects > [objectname] Template Category** and click **Next** to open the **Create New [objectname] Template Category** wizard.
2. Specify a name for the template category in the **Name** field.
3. Optionally enter a comment to describe the template category in the **Comment** field.
4. Click **Finish** to close the wizard.

The template category that you created for a visual project is added to the appropriate **[objectname] Templates** folder under the **Templates** node in the **Repository Explorer** view.

Note:

The visual application template categories that were created in previous versions of QMF are contained in the **Visual Application Templates** folder under the **Templates** node. You can edit them, and then save them as visual dashboards to a repository or file. You cannot add new visual application templates in QMF.

The **Visual Application Templates** folder is not supported in the new repositories that you create.

Creating visual reports

Visual reports are page-based reports that include both text and graphics.

About this task

Using QMF for Workstation, you can quickly design visual reports using the intuitive **Visual Designer** perspective that includes an editor that presents both design and runtime views of the report; the **Project Explorer** view which details the structural content of each visual report; and the **Palette, Properties, and Output** views which support the Visual Designer editor and assist in creating visual reports.

Note: You can generate and view previously designed visual reports in QMF for WebSphere, but you cannot design visual reports. The **Visual Designer** perspective is not available.

The following steps outline the tasks that you will perform when creating or editing visual reports:

Procedure

1. Activate a visual report in the editor window. You can activate a visual report in the workstation editor window in one of the following ways:
 - Create a new visual report from a set of query results using the **Display Report** wizard.
 - Create a new visual report using the **New Visual Report** wizard.
 - Open an existing visual report.
 - Import a legacy visual report.
2. Open the **Visual Designer** perspective. The **Visual Designer** perspective presents an editor window, several views and the menus that you will use when you work with visual reports. You can open the **Visual Designer** perspective by using either the **Open perspective** toolbar button or by selecting **Window > Open Perspective > Visual Designer**.
3. View the key components of the visual report as listed in the **Project Explorer** view. Each visual report is listed under the **Visual Reports** node in the **Project Explorer** tree. Each visual report will have the following folders:
 - Connections
 - Queries
 - Globals
 - Main Page
 - Fixed Pages
4. Create data source connections. Visual reports are intended to be shared across the enterprise. In order to facilitate sharing and distributing visual reports, QMF for Workstation does not tie the queries that are included in a visual report to a specific data source. Instead it ties the query to a connection information alias. You must set up a connection information alias for each data source that will be used to obtain data for the visual report. All connection information aliases that are available for a report are listed in the **Connections** folder for the report in the **Project Explorer** view. You can set up a connection information alias for a data source using the **Insert Connection** dialog or by dragging the data source from the **Workspaces** view to the **Connections** folder for the visual report in the **Project Explorer** view.
5. Specify the main query that will supply the data for the visual report. You can specify an existing query or create a new query. When you specify the query you will be asked to associate the query with a

connection information alias. The connection information alias will identify the data source against which the query will be run. The main query for a report is listed in the **Queries** folder for the visual report in the **Project Explorer** view. You can specify the query using the **Insert Query** dialog or by dragging the query from the **Workspaces** view to the **Queries** folder for the visual report in the **Project Explorer** view.

6. Design the main page of the report. The main query provides the data for the report and this data is formatted based on the design of the main page. See [“Designing the main page of a visual report” on page 343](#) for more information on designing the main page.
7. Design one or more fixed pages that will be included in the report. Fixed pages are single pages with unique elements that can be inserted in the visual reports. See [“Designing a fixed page for a visual report” on page 343](#) for more information on designing fixed pages.
8. Each time that you click the **Preview** button you are actually running the visual report object and you can view how your results will appear.
9. When you have completed designing or editing the visual report, you can save the report, print the report, or export the report to a HTML or PDF file.

Restriction: In 64-bit version of the application, when visual reports are run in the PDF mode, they are opened in the third-party application.

Creating reports using the visual report wizard

The **Create New Visual Report** wizard will help you create a new visual report.

About this task

To create a new visual report using the **Create New Visual Report** wizard:

Procedure

1. Open the **Create New Visual Report** wizard in one of the following ways:
 - From the **Visual Designer** perspective, right-click the **Visual Reports** folder in the **Project Explorer** view. Select **New Visual Report** from the pop-up menu. The **Create New Visual Report** wizard opens.
 - From the **Visual Designer** perspective, select **File > New > Visual Report**. The **Create New Visual Report** wizard opens.
 - With an active query in the editor window, select **Query > Transfer To > Visual Report**.
 - Click the **New Visual Report** toolbar button.
2. Specify the name of the new visual report in the **Visual report name** field.
3. Select the template that will be used for the new visual report in one of the following ways:
 - Select the **Predefined** tab. From the list of predefined templates, select the one on which you want to model the new visual report. Predefined templates are delivered with the application. The layout of predefined templates cannot be changed. The **Blank** predefined template is the default template for all visual reports. It contains no structural elements.
 - Select the **Local** tab. From the list of local templates select the one on which you want to model the new visual report. Local templates are those that have been created for your organization and saved locally. Each template contains predefined content (such as company logos, standardized text, frequent graphical layouts). The templates are listed by category for organizational purposes. Users can edit the structural content of local templates to suit their needs.

Note: The **Local** tab is not displayed if there is no locally saved templates in your Templates directory.
 - Select the **Repository** tab. From the list of templates that are saved in the repository select the one on which you want to model the new visual report. Templates listed on the repositories page are those that have been created for your organization and saved in the repository. Each template contains predefined content (such as company logos, standardized text, frequent graphical layouts).

The templates are listed by category for organizational purposes. Users can edit the structural content of repository templates to suit their needs.

4. Click **Finish**. A new visual report object opens in the editor. The design template for the main page is displayed. The **Project Explorer** view opens (if it is not already open) and lists the new visual report under the **Visual Reports** node in the tree.

Creating a new visual report template

Users can create visual reports with predefined content, such as company logos, standardized text, and special graphics. Each visual report can then be saved as a template that can be used by others to facilitate easy authoring of visual reports.

About this task

To create a new visual report template:

Procedure

1. Create a new visual report and include the common elements or open an existing visual report that will serve as the basis of the template.
2. Select **File > Save As**. The **Save** wizard opens.
3. Specify where you will save the visual report template. Select **Save to Repository** to save the report in a repository. Select **Save to File** to save the visual report locally.
4. Select the **Save as template** check box to save the visual report as a template.
5. Click **Next**. The second page of the wizard varies depending on where you are saving the visual report template.

If you are saving the template locally, the **Enter file name or choose file within a file dialog** page of the wizard opens. You will specify the path to your local directory where you will save the template in the **Path** field. For example, in a Windows operating system the path would be:

- For Windows XP or earlier:

```
C:\Documents and Settings\[UserName]\Application Data\IBM\
QMF for Workstation\Templates\VR Templates\VisualReport1
```

- For Windows Vista or later:

```
C:\Users\[UserName]\Application Data\IBM\
QMF for Workstation\Templates\VR Templates\VisualReport1
```

You can replace `VisualReport1` with any name. It is recommended that the path be to the local directory as this is where QMF for Workstation will look for all local templates in order to display them to users when they create a new visual report. You can use the **Browse** button to search for a location.

6. If you are saving the template to a repository, the **Set up the repository object parameters** page of the wizard opens. The **Select parent entry** list displays a list of categories that you have set up to organize your repository templates. If you have not set up any categories, this list is blank. Click the **New Visual Report Template Categories** icon to create a category. Specify a name for the report in the **Name** field. Optionally you can specify a comment in the **Comment** field.
7. Click the **New Visual Report Template Categories** icon to create a category.
The Create New Visual Report Template Category wizard opens.
8. Specify a name for the template category. Optionally you can specify a comment in the **Comment** field.
9. Click **Finish**.
The new visual report template category has been created. It is displayed in the Select parent entry list.

10. Specify a name for the report in the **Name** field. Optionally you can specify a comment in the **Comment** field.
11. Click **Finish**. The new visual report template is saved. It will be presented to users when they create a new visual report.

Opening a legacy visual report

Legacy visual reports are visual reports that were created using versions of QMF applications prior to 9.1.

About this task

You can open legacy reports directly or by importing the report. Once opened in QMF for Workstation V11.1 (or later), these reports can only be saved in a repository. They cannot be saved in a QMF catalog because once opened in V11.1 they cannot be opened in previous versions of QMF applications.

The following tasks describe opening legacy visual reports:

Opening legacy visual reports directly

You can open legacy reports directly.

About this task

To open a legacy visual report directly:

Procedure

1. Select **File > Open From**. From the pop-up menu select one of the following options depending on where the visual report object resides:
 - **QMF catalog**
 - **Repository**
 - **File**
2. One of the following three windows opens depending on your selection:
 - If you selected **QMF catalog**, the **Open from QMF Catalog** window opens. In the **Data Source** field specify the name of the data source where the QMF catalog resides. In the **Owner** field specify the owner of the object that you want to open. In the **Name** field specify the name of the object that you want to open. Select the visual report object. The visual report opens in the editor window. Select the **Run Immediately** check box to run the visual report when it is opened. The visual report opens in the editor window. If you selected the **Run Immediately** check box the result set is displayed.
Note: You can use the **Object List** dialog to find a legacy visual report. For this, click **From List**. See [Listing data source objects](#) for more information.
 - If you selected **Repository**, the **Open from Repository** window opens. The contents of the current repository are displayed in a tree structure in the **Select repository entry(s)** list box. Expand the folders until you uncover the legacy visual report object that you want to open. Select the visual report object. The visual report opens in the editor window.
 - If you selected **File**, an operating system specific **Open** window opens. Type the full path name for the file in the file name field. You can use the browse function to search for the location of the file. Select the visual report object. The visual report opens in the editor window.
3. The legacy visual report opens. It can no longer be opened in a version of QMF prior to V9.1.

Importing legacy visual reports

You can import legacy visual reports.

About this task

To import a legacy visual report:

Procedure

1. From any perspective, select **File > Import**. The **Import** window opens.
2. Select **QMF Visual Report File**.
The **QMF Visual Report Import** Wizard opens.
3. Click **Next**. The **QMF Visual Report Import** Wizard opens.
4. In the **File path** field, specify the path to the visual report file that you want to import. You can use the **Browse** button to search for the file.
5. Type a unique name for your new visual report that will be created from the legacy visual report in the **Report name** field.
6. Type the location where you want to save the new visual report (within the currently connected repository) in the **Location** field.
7. Click **Finish**. A folder with the visual report name is added to the **Visual Reports** node in the **Project Explorer** view. For each new visual report, folders named **Connections**, **Queries**, **Globals**, and **Scenes** are also added in the **Project Explorer** tree. The editor window opens with a canvas where you can make edits to the converted report.

Specifying data source connections for visual projects

You set up a connection information alias for each data source that will provide data for the visual project (visual report or dashboard).

About this task

When you add the queries that will be used in a visual project, you must specify what connection information alias will be associated with each query.

To change the data source that will be used for a query, you edit the query information and select a connection information alias entry that points to the data source that you want to use. This feature allows users to easily share visual reports and use their own data sources. In addition, it facilitates moving between test and production data sources.

To specify a connection information alias for a data source, perform the following steps:

Procedure

1. Open the **Insert Connection** window.
You can open the **Insert Connection** window in one of the following ways:
 - From the **Project Explorer** view, right-click the **Connections** node for the specific visual project. Select **Insert Connection** from the pop-up menu.
 - From the **Project Explorer** view, double-click the **Connections** node. The **Insert Connection** window opens.
 - Click a query from the **Workspaces** or **Repository Explorer** view. With the mouse button pressed, drag the query to the **Queries** node in the **Project Explorer** view. A new data source connection is automatically added to the **Connections** folder of the **Project Explorer** view.
2. From the list of available data sources, select the data source that will be associated with this connection information alias.
3. Specify a unique name for this connection information alias in the **Connection name** field.
4. Click **Finish**. The **Project Explorer** view lists the new connection information alias under the **Connections** node in the tree.

Note: You can also set up a connection information alias for a data source by dragging the data source from the **Workspaces** view to the **Connections** folder for the visual project in the **Project Explorer** view. The connection information alias is added with the name **ConnectionN**. You can edit the name of the connection information alias by right-clicking the entry and selecting **Rename**.

Results

You have set up the data source connection information for the visual data object.

Specifying queries for visual projects

Visual projects (visual reports or dashboards) can include one or more queries to run to obtain the result data and display it in the visual project.

About this task

You can create a new query that will be included in the visual project or you can use existing queries.

To specify a query for a visual project, perform the following steps:

Procedure

1. From the **Project Explorer** view, right-click the **Queries** node and select **Insert Query** from the menu.
2. Specify a unique name for the query in the **Query name** field.
3. Identify the query and specify where it resides in one of the following ways:
 - a) If you are creating a new visual query, select **Create a new visual query**.
 - b) If you are creating a new analytical query, click **Create a new analytical query**.
 - c) If you are creating a new forecast, click **Create a new forecast**.
 - d) If the query exists, select **Attach to an existing query**. The **Select query** list appears. Specify the name of the query and where it resides in the **Select Query** field. You can click **Browse** to search for the query.
 - e) If you would like to insert a static query, select **From file**.
 - Select **Insert Data** to copy the query directly into the **Queries** folder. This option provides greater portability, because the query will stay with the visual project if it is moved from one system to another.
 - Select **Use link to file** to create a link to the query's system directory location. This option limits portability, because if the visual project is moved from one system to another, the query will no longer be available.
 - Specify the query to be added in the **Select data file** list. You can click **Browse** to search for the query.
4. If you insert , you must select a data source connection that you want to use for the query from the **Connection** list. If the **Connection** list is empty, add a data source connection to a visual project.
5. Click **Add Connection** to open the **Insert Connection** window.
6. From the data sources tree, select the data source that you want to add for the visual project and click **Finish**.

The data source connection that you selected is added to the visual project and displayed in the **Connection** list.
7. If you are inserting an existing forecast or analytical query into a visual project, you can edit connection settings for subordinate queries in the **Edit Connections** window.
8. If you are inserting a new forecast or analytical query you cannot set connections for subordinate queries because these queries are not specified yet.
9. Click **Finish**. The **Project Explorer** view lists the query under the **Queries** node in the tree. If the query requires a parameter, a **Parameters** folder is added. You will automatically save the query when you save the visual project.

Results

Note: You can also add a query by dragging the query from the **Workspaces** view to the **Queries** node of the visual project in the view. The selected query is added to the **Queries** folder. A new data source connection is automatically added to the **Connections** folder of the **Project Explorer** view.

Designing the main page of a visual report

The main page of a visual report is where you specify the content for the overall report. You will specify the header, detail, and footer template information that will be repeated for each page of the report.

About this task

A visual report main page is divided into the following sections:

- **Report Header** section describes what appears at the beginning of the report.
- **Page Header** section describes what appears at the top of each page of the report.
- **Detail Header** section describes what appears as column headers to each detail line.
- **Detail** section describes how each detail line of data will be presented.
- **Detail Footer** section describes what appears on the break line for a detail section.
- **Page Footer** section describes what appears at the bottom of each page of the report.
- **Report Footer** section describes what appears at the end of the report.

The process of designing each section of the main page of a visual report is the same for each section. To specify what should appear in each section of the visual report:

Procedure

1. You can select which section you will be designing in one of the following ways:
 - From the **Project Explorer** tree, select the folder for the section of the report that you want to design.
 - In Design mode, the editor window is divided into multiple sections each labeled and associated with a section of the report. Click in the section that you want to design.
2. In each section of the report, you can insert design elements that identify what will be included in that section of the report. You choose the design elements from the **Palette** view. Select the objects that you want to insert. Then click in the section of the report where you want to place the object.
3. The properties of each section of the visual report and the properties of each object that you can insert in the report are displayed in the **Properties** view. You can modify the properties of each section of the visual report and the properties of each object in the **Properties** view. The properties of some of the report sections and objects can also be changed in the editor window.
4. You can view your report at any time during the design phase. Click the **Preview** tab in the editor window to display the report in preview mode. To return to the design mode, select the **Design** tab.
5. You can save the report to a file, in the QMF catalog if it is available, or in a repository. You can also print the report or save it as an HTML or a PDF file.

Designing a fixed page for a visual report

Fixed pages of a visual report are single pages with unique elements that can be inserted in the visual report. Fixed pages can be used for example as cover pages, chapter beginning pages, special graphic pages, or a report closing page.

About this task

A visual report fixed page is divided into the following sections:

- **PageHeading** section describes what appears at the top of the fixed page.
- **Fixed1** section describes what appears in the first fixed section of the page.

- **Fixed2** section describes what appears in the second fixed section of the page.
- **PageFooting** section describes what appears at the bottom of the fixed page.

The process of designing each section of a fixed page is the same for each section. You can create as many fixed pages as your report requires. You specify where the pages will appear in the report using the property **FixedPages** that is available for each of the main page sections.

To create a fixed page for your visual report:

Procedure

1. When you create a new visual report object, you automatically get a folder in the **Project Explorer** that can be used to design your first fixed page. You can however create as many fixed pages as is required by your report. To create additional fixed pages, right-click the **Fixed Pages** node from the **Project Explorer** view. Select **Insert Fixed Page** from the pop-up menu. The **New Fixed Page** wizard opens. Specify a unique name for this fixed page in the **Fixed page** name field. Click **Finish**. The **Project Explorer** view lists the new fixed page under the **Fixed Pages** node in the tree.
2. Select which section you will be designing in one of the following ways:
 - From the **Project Explorer** tree, click the folder for the section of the fixed page that you want to design.
 - In the Design mode, the editor window is divided into multiple sections each labeled and associated with a section of the fixed page. Click the section that you want to design.
3. In each section of the fixed page, you can insert design elements that identify what will be included in that section of the page. You choose the design elements from the **Palette** view. Select the objects that you want to insert. Then click the section of the fixed page where you want to place the object.
4. The properties of each section of the fixed page and the properties of each object that you can insert in the report are displayed in the **Properties** view. You can modify the properties of each section of the fixed page and the properties of each object in the **Properties** view. The properties of some of the fixed page sections and objects can also be changed in the editor window.
5. Specify where to insert the fixed page in the main report using the property **FixedPages** that is available for each of the main page sections. If you specify a **FixedPages** property in the:
 - Report header section: the fixed page will appear as the cover page of the report.
 - Page header section: the fixed page will appear before the highest level break.
 - Detail header section: the fixed page will appear before the detail level break.
 - Detail footer section: the fixed page will appear before the detail level break.
 - Page footer section: the fixed page will appear after the highest level break.
 - Report footer section: the fixed page will appear as the last page of the report.
6. You can view your fixed page at any time during the design phase. Click the **Preview** tab in the editor window to display the report in preview mode. To return to the Canvas mode, select the **Design** tab.

Managing report data retrieval

You can control when data is retrieved for visual reports.

About this task

By default, visual reports are run against live data. Each time the visual report is run, queries are run, and the data in the report is updated. You control the data retrieval for a visual report using a set of expiration schedules. This option allows you to specify when, if at all, you want to run the visual report queries to update the data that is displayed in the report.

To set up expiration schedules for a visual report's data retrieval:

Procedure

1. Create a new visual report, including the queries that will be used to retrieve data.
2. From the **Project Explorer** view, expand the **Globals** folder, and right-click the **Expiration Schedules** node for the visual report. Select **Insert Schedule** from the pop-up menu. The **Expiration Schedule Configuration** wizard opens.
3. You can create one or more schedules that will control the data retrieval for the visual report. Specify the name of the schedule in the **Schedule name** field. This can be any name that has meaning.
4. The type of schedule controls when data that has been retrieved for this visual report will expire. You can choose one of the following schedule types by selecting one of the **Data expires** radio buttons:
 - **daily**: Select this option to specify that the same data will be used for a 24 hour period. If you select this option, the visual report will run, data will be retrieved, and every time the report is run in the next 24 hour period it will use the same data. When the 24 hour period expires, the next time the report is run, new data will be retrieved.
 - **weekly**: Select this option to specify that the same data will be used for a period of 7 days. If you select this option, the visual report will run, data will be retrieved, and every time the report is run in the next 7 days it will use the same data. When the 7 day period expires, the next time the report is run, new data will be retrieved.
 - **monthly**: Select this option to specify that the same data will be used for a month. If you select this option, the visual report will run, data will be retrieved, and every time the report is run in the next month it will use the same data. When the month expires, the next time the report is run, new data will be retrieved.
 - **at specific date**: Select this option to specify that the same data will be used until a specific date is reached. If you select this option, the visual report will run, data will be retrieved, and every time the report is run up until the specific date it will use the same data. When the specific date is reached, the next time the report is run, new data will be retrieved.
 - **never expires**: Select this option to specify that the same data will always be used. If you select this option, the visual report will run, data will be retrieved, and every time the report is run up it will use the same data.
 - **always expires**: Select this option to specify that the data will always be updated every time the report is run.
 - **after the visual project is closed** to have the schedule expire after you close the visual project.
5. If you selected **daily**, **weekly**, **monthly**, or **at specific date**, click the **Next**.
6. Depending on the type of schedule that you are creating one of the following variations of the **Set schedule properties** page of the wizard opens:
 - If you selected **daily**, you will specify the date that the 24 hour period will start in the **Begin date** field and the time the 24 period will start in the **Begin time** field. You specify when the 24 hour period will occur using the **Data expires** radio buttons. Select **daily** to specify that the 24 period will occur every day. Select **each** to specify that the 24 hour period will occur every number of days. Select the number from the drop-down list. Select **days of week** to specify that the 24 hour period will occur on one or more specific days. You must select the check boxes of the days.
 - If you selected **weekly**, you will specify the date that the week will start in the **Begin date** field and the time the week will start in the **Begin time** field. You can select a number from the **each week** list to specify that the 7 day period will occur every number of weeks. You can select a day from the **days of week** check boxes to specify that your 7 day period will begin on a specific day of the week.
 - If you selected **monthly**, you will specify the date that the month will start in the **Begin date** field and the time the month will start in the **Begin time** field. You can select **or** to specify that data retrieval will occur on a specific day of the week in the month. Select the month from the select months list of check boxes.
 - If you selected **at specific date**, you will specify the date and time that will trigger data retrieval in the **Date and time** list box. Click the cell to bring up a calendar where you will select the date and time.

7. Click **Finish**. The schedule is created and listed in the **Project Explorer** view. You will repeat this process to define additional properties. After all schedules are defined you will assign the schedules to the queries in the visual report.

Assigning data retrieval schedules to visual report queries

To control visual report data retrieval, you must assign the data retrieval schedules to the queries in the visual report.

About this task

To assign data retrieval schedules to visual report queries:

Procedure

1. Right-click the **Queries** node for the visual report in the **Project Explorer** view. Select **Assign Expiration Schedules** from the pop-up menu. The **Manage expiration schedules** wizard opens.
2. Select **Use expiration schedules** to assign an expiration schedule to the visual report.
3. Select the default expiration schedule from the **Default expiration schedule** list. This schedule will be used to control the data retrieval for all queries that are included in the visual report unless they have been assigned a specific schedule.
4. You can assign specific schedules to specific queries. Each query that is included in the visual report is listed in the **Query name** cell. To assign a specific schedule to the query, click in the corresponding **Schedule name** cell. Select the unique schedule from the drop-down list of available schedules.
5. Click **Refresh cache** to refresh the retrieved data for the query.
6. Click **Finish**. Any schedule assignments that you have made are saved.

Converting visual reports to offline reports

You can convert a visual report to an offline report. An offline report is a static version of a visual report. Users can open the report and will always see the same data.

About this task

To create an offline report from a visual report:

Procedure

1. Open a visual report.
2. Select **Edit > Convert to Offline Report**. All the queries for the visual report will be run and all the data that is needed to render the report will be retrieved.
3. Save the offline report in the repository. You can not save the report in a file.
4. To convert an offline report back to an online report, open the offline report and select **Edit > Convert to Online Report**.

Exporting reports to a file

You can export a classic report, quick report, or a visual report to a file. If your report is a classic one, you can export it as a text, HTML or PDF file. If your report is a quick one or a visual one, you can export it as an HTML or a PDF file.

About this task

To export a report:

Procedure

1. Open the **Export Report** wizard in one of the following ways:

- With a report in the active window, select **File > Export** to open the **Select** page of the **Export** wizard, and then select **Report** from the list of available export wizards.
 - With a report in the active window, click the **Export** toolbar button.
2. Select the report that you want to export from the **Reports** list box.
 3. If you are using QMF for Workstation, specify where the report file should be saved in the **Report file** field. You can use the **Browse** button to search for the location.
 4. Select the type of file that you are exporting. You can export a classic report as a text file (*.txt), an HTML file (*.htm;*.html), or a PDF file (*.pdf). You can export a quick report and a visual report as an HTML file (*.htm;*.html) or a PDF file (*.pdf).
 5. If you are exporting a PDF file, click the **Next** button. The **Set up PDF export options page** opens. Select the **Add Document Information** check box and specify character strings in the **Title**, **Author**, **Subject** and **Keywords** fields to add document information to the PDF. Select the **Use only ANSI fonts** check box to use only ANSI fonts in the PDF.

Note: Some of the fonts (for example, TTC) can be displayed incorrectly in PDF files.
 6. If you want to export your report as an HTML file, select HTML from the **Export type** list and click **Next**. The **Set up file export options** page opens. Select the **Split report into separate pages** check box to save each page of you report individually.
 7. Click **Finish**.
 - If you are using QMF for Workstation, the report is exported to a file in the format that you selected.
 - If you are using QMF for WebSphere, a **Save** window opens that list the individual files for download, and an archived ZIP file (*.zip) that contains all of the files for download. Click one or more of the hyper-links to download the files. When you have finished downloading the files, click **OK**.

Working with Ad hoc reports

Ad hoc reports are Canvas-based reports that graphically represent information retrieved from a database. An Ad hoc report includes a scene with a Canvas.

You can place multiple objects on the Canvas and manage the Canvas space to suit your purposes.

The Canvas Internal Tree displays the objects in all available workspaces and allows you to add them to your report.

The Canvas supports charts and grids that can be based on the following objects:

- Analytical Query
- Compound Dynamart
- Drill-down Path
- Dynamart
- Table
- Visual Query

The Canvas with all the objects is saved as a Built-in layout when ad hoc report is saved. You can create multiple layouts for each report.

You can also link objects in the Canvas of ad hoc reports and visual dashboards.

Creating Ad hoc reports

You can create an Ad hoc report and populate it with objects in the Ad hoc Reports runtime editor.

About this task

An Ad hoc report includes a scene with a canvas. When creating an Ad hoc report, you can place multiple objects on the Canvas and manage the Canvas space according to your purposes.

To create an Ad hoc report perform the following procedure:

Procedure

1. From the main menu, select **File > New > Other**.
The **New** window opens.
2. Expand **QMF Objects** and select **Ad hoc Report**. Click **Next**.
3. In the **Name** field, specify the unique name for your Ad hoc report. Click **Finish**.

Results

You created an Ad hoc report. Now, you must add objects to it.

Related concepts

[“Managing Canvas operations using the Canvas toolbar” on page 435](#)

The Canvas toolbar provides actions for managing the Canvas in ad hoc reports and visual dashboards.

Related tasks

[“Wiring objects on the Canvas” on page 436](#)

Wiring objects defines the relationship and dependencies between objects. Use the object Wiring Mode to connect objects on the Canvas.

Working with visual dashboards

A Visual dashboard is a type of visual project that presents interactive or persistent data to a wide range of users.

About this task

Visual dashboards can simultaneously draw data from multiple, heterogeneous data sources and present the data using a variety of graphs, charts, and graphical widgets. Using QMF for Workstation, content developers can create a visual dashboard that can be viewed by multiple users with either QMF for Workstation or QMF for WebSphere.

Note: You can view previously designed visual dashboards in QMF for WebSphere, but you can not design them. The **Visual Designer** perspective is not available.

The following steps outline the tasks that you will perform when creating or editing visual dashboards:

Procedure

1. Activate a visual dashboard in the workstation editor window. You can activate a visual dashboard in the workstation editor window in one of the following ways:
 - Create a new visual dashboard using the **Create New Visual Dashboard** wizard.
 - Open an existing visual dashboard.
2. Open the **Visual Designer** perspective. The **Visual Designer** perspective presents an editor window and several views and menus that you will use when you work with visual dashboards. You can open the **Visual Designer** perspective by using either the **Open perspective** toolbar button or by selecting **Window > Open Perspective > Visual Designer**.
3. View the key components of the visual dashboard as listed in the **Project Explorer** view. Each visual dashboard is listed under the **Visual Dashboards** node in the **Project Explorer** tree. Each visual dashboard will have the following folders:
 - **Connections**
 - **Globals**
 - **Queries**
 - **Scenes**
4. Create data source connections.

Visual dashboards are intended to be shared across the enterprise. In order to facilitate sharing and distributing visual dashboards, QMF for Workstation does not tie the queries that are included in a visual dashboard to a specific data source. Instead, it ties the query to a connection information alias.

You must set up a connection information alias for each data source used to obtain data for the visual dashboard. For information about setting up a connection information alias, see [Specifying data source connections for visual projects](#).

5. Specify the first query that will supply data for the visual dashboard.

You can specify an existing query or create a new query. When you specify the query the system prompts you to associate the query with a connection information alias. The connection information alias identifies the data source against which the query runs. For information about setting up queries, see [“Specifying queries for visual projects”](#) on page 342.

6. Design the first scene for the dashboard. A visual dashboard consists of one or more scenes.

A scene is the container (similar to a presentation slide) that will hold all the elements used to display your data. After creating a new visual dashboard, the system provides one default scene. You can choose to have only one scene in your visual dashboard or you can add additional scenes.

7. Insert objects into a scene.

The objects that you can insert are displayed in the **Palette** view and are organized on individual palettes based on their type. From the **Palette** view select the objects that you want to insert into your scenes.

8. Modify the object's properties in the **Properties** view.

By modifying an object's properties you set the values that determine how the object looks and behaves. You can use the **Expression Designer** to help you specify an object's properties.

9. Insert the layout objects

In a visual dashboard scene layout objects display query results.

When you insert a layout object the system prompts you to associate the object with a query that is contained in the **Queries** folder. You can add queries to your **Queries** folder at any time during the process of creating your dashboard.

Layout objects can present data in many ways. For most layout objects, you display the results of multiple queries in a single layout. For example, you might create a single XY chart that displays sales figures derived from one query and spending figures derived from another query. You can add multiple layout objects. When you place layout objects, you can pass query result information from a higher level layout object to a lower level layout object. Because you have the ability to pass this information, you can use the placed layout objects to display more detailed information that relates to a specific data value.

10. Use the **Events** view to assign the different navigation options.

By assigning navigation options to scenes and objects, you set a path through the dashboard. One form of navigation is to define dependencies between dashboard control objects (such as combo or list box objects) and data-driven layout objects (such as a grid or bar chart object) using the **Connectivity** tool from the **Palette** view.

11. Run the visual dashboard.

Each time that you click the **Runtime** tab, you are actually running the visual dashboard object. By performing this step you can review how the results will display at run time. Think of this step as a test-run of your dashboard that allows you to improve upon the design before saving and deploying the dashboard.

12. Save the visual dashboard.

When you have completed designing or editing the visual dashboard, you can save the dashboard and copy it to one or more workspaces for distribution to other users.

Planning visual dashboards

Before developing a visual dashboard, plan your development and design process. One of the most important planning activities for designing visual dashboards is known as *storyboarding*.

Using storyboards

Storyboards are visual aids to a design and a design process. Acting as a graphical organizer, storyboards help all members of your team pre-visualize the dashboard that you intend to design.

The information in this section is intended as a general guideline to using storyboards.

Proper use of storyboards during the planning phase facilitates the design process by clearly illustrating how the dashboard will look and function.

Storyboards help take the guesswork out of the actual design phase and can minimize potential development missteps and help you deploy the dashboard more efficiently. Additionally, a well designed dashboard has a higher likelihood for reuse in other areas of the enterprise.

You use a storyboard to design the basic features, functionality and presentation flow of the scenes and objects to be included in your visual dashboard.

A storyboard for a visual dashboard should specify the following:

- Data for each scene
- Scenes, drawn roughly
- Layouts for each scene
- Sequence of presentation
- Points where users can access further information
- Actions that result from user events
- Jumps between scenes

There are many types of tools, with varying level of complexity, that you can use to create storyboards. You can use a pencil and paper to sketch the design and flow. You can use a presentation tool, such as Microsoft PowerPoint, to create sample layouts, with supporting detail on notes pages. You can also put together Web pages to show actual jumps and navigation flow.

Use the following list of general topics as a guide to elements you should include in your storyboard / design:

Displaying query results data

You use layout objects or the List Box and Combo Box controls to display query results data. Explore the different display options and decide which ones best suit the kind of analysis you want to show.

Capturing user input

You can capture user preferences with standard user interface controls. Look at the **Controls** palette on the **Palette** view to explore your options.

Passing information

Using parameters, you can pass information acquired from user actions to affect the display of a dashboard, the contents of a scene, or the execution of a query.

Navigating around dashboards

You can use navigation features to allow users to move around dashboards to get to new information.

Locating your data

Your storyboard should specify the database and include a list of the tables that you plan to use. Writing this list will help you organize your work. You can then spend a session creating all the queries you need prior to creating your first scene. Alternatively, you might work with others who extract the data that you need from the database tables and consolidate it in summary tables.


Creating Visual Dashboards using the Visual Dashboard wizard

You can create a new Visual Dashboard using the **Create New Visual Dashboard** wizard.

About this task

To create a new Visual Dashboard, perform the following steps:

Procedure

1. Open the **Create New Visual Dashboard** wizard on one of the following ways:
 - From the **Visual Designer** perspective, select **File > New > Visual Dashboard**. The **Create New Visual Dashboard** wizard opens.
 - From the **Visual Designer** perspective, right-click the **Visual Dashboards** folder in the **Project Explorer** view. Select **New Visual Dashboard** from the pop-up menu. The **Create New Visual Dashboard** wizard opens.
 - From the **Visual Designer** perspective, double-click the **Visual Dashboards** folder in the **Project Explorer** view. The **Create New Visual Dashboard** wizard opens.
 - Click the **New Visual Dashboard** toolbar button().
2. Type a unique name for your dashboard in the **Dashboard name** field.
3. Select the template that will be used for the new visual dashboard in one of the following ways:
 - Select the **Predefined** tab. From the list of predefined templates select the one on which you want to model the new visual dashboard. Predefined templates are delivered with the application. The layout of predefined templates cannot be changed. The **Blank** predefined template is the default template for all visual dashboards. It contains no structural elements.
 - Select the **Local** tab. From the list of local templates select the one on which you want to model the new visual dashboard. Local templates are those that have been created for your organization and saved locally. Each template contains predefined content (such as company logos, standardized text, frequent graphical layouts). The templates are listed by category for organizational purposes. Users can edit the structural content of local templates to suit their needs.

Note: The Local tab displays only if there are locally saved templates in your Templates directory.
 - Select **Repository** tab. From the list of templates that are saved in the repository select the one on which you want to model the new visual dashboard. Templates listed on the repositories page are those that have been created for your organization and saved in the repository. Each template contains predefined content (such as company logos, standardized text, frequent graphical layouts). The templates are listed by category for organizational purposes. Users can edit the structural content of repository templates to suit their needs.
4. Click **Finish**. A folder with the dashboard name is added to the **Dashboards** node in the **Project Explorer** view.

For each new dashboard a folder is created for the **Connections**, **Queries**, and **Globals**. In addition, a **Scenes** folder is added with a default first scene. The editor window opens with a canvas where you will design the first scene of the dashboard.

Results

You have created a new Visual Dashboard using the **Create New Visual Dashboard** wizard.

What to do next

You are now ready to design your dashboard, adding **Connections**, **Queries** and **Globals** and additional **Scenes**.

Creating a new visual dashboard template

Users can create visual dashboards with predefined content, such as company logos, standardized text, and special graphics. Each visual dashboard can then be saved as a template that can be used by others to facilitate easy authoring of visual dashboards.

About this task

To create a new visual dashboard template:

Procedure

1. Select **File > Save As**. The **Select a wizard** window opens.

From the **Select a wizard** window, select which wizard to use, select the **Save as template** check box and then click **Next**:

- Save to File

Select this option if you want to use the **Save to File** wizard to save the Visual Dashboard locally to a file.

See step “2” on page 352.

- Save to Repository

Select this option if you want to use the **Save to Repository** wizard to save the Visual Dashboard to a repository.

See step “3” on page 352.

2. On the **Save to File** page, enter the file location in the **Path:** field or select **Browse. .** to locate the file and click **Finish**.

For example, in a Windows operating system the path would be one of the following:

- For Windows XP or earlier:

```
C:\Documents and Settings\[user_name]\Application Data\IBM\
QMF for Workstation\Templates\VD Templates\VisualDashboard1
```

- For Windows Vista or later:

```
C:\Users\[user_name]\Application Data\IBM\
QMF for Workstation\Templates\ VD Templates\VisualDashboard1
```

You can replace `VisualDashboard1` with any name. It is recommended that the path be to the local directory as this is where QMF for Workstation will look for all local templates in order to display them to users when they create a new Visual Dashboard.

You have saved the Visual Dashboard template to a file.

3. On the **Save to Repository** page, set up the repository object parameters.

The **Select parent entry** list displays a list of categories that you have set up to organize your repository templates.

If you have not set up any categories this list is blank.

4. *Conditional:* Click the **Create New Visual Dashboard Template Categories** icon to create a category.

The **Create New Visual Dashboard Template Category** wizard opens.

5. Specify a name for the template category. Optionally you can set a comment in the **Comment** field.
6. Click **Finish**.

The new Visual Dashboard template category has been created. It is displayed in the **Select parent entry** list.

7. Specify a name for the dashboard in the **Name** field. Optionally you can set a comment in the **Comment** field.
8. Click **Finish**.

The new Visual Dashboard template is saved. It will be presented to users when they create a new visual dashboard.

Importing a legacy Visionary world

You can create a new dashboard from an existing Visionary world that was created using a previous version of QMF Visionary.

About this task

The Visionary world that you import must have been exported as an XML file using QMF for Visionary Version 8.1, Fix Pack 13 or later. To export a Visionary world using QMF for Visionary Version 8.1, Fix Pack 13, you select **File > Export as XML**

To import a legacy Visionary world:

Procedure

1. From any perspective, select **File > Import**. The **Import** window opens.
2. Select **Visionary World File**.
3. Click **Next**. The **Visionary World Import** wizard opens.
4. In the **File path** field, specify the path to the Visionary world that you want to import. You can use the browse button to search for the file.
5. Type a unique name for the new dashboard that will be created from the legacy Visionary world in the **Dashboard name** field.
6. Type the location where you want to save the new dashboard (within the currently connected to repository) in the **Location** field.
7. Click **Finish**. A folder with the dashboard name is added to the **Dashboards** node in the **Project Explorer** view. For each new dashboard, folders named **Connections**, **Queries**, **Globals**, and **Scenes** are added. The editor window opens with a canvas where you can make edits to any scene within the converted world.

Specifying data source connections for visual dashboards

You set up the connection information alias for each data source that will be used in the visual dashboard.

About this task

Visual dashboards are intended to be shared across the enterprise. In order to facilitate sharing and distributing visual dashboards, QMF for Workstation does not tie the queries that are included in a visual dashboard to a specific data source. Instead, it ties the query to a connection information alias. The connection information is saved with the visual dashboard and can be easily modified to point to any data source.

The connection information includes a name for the connection and the name of the data source to which it will point. When you add the queries that will be used in the visual dashboard, you will specify what connection information alias will be associated with the query.

To change the data source that will be used for a query, you edit the connection information and point to the data source that you want to use. This feature allows users to easily share visual dashboards and use their own data sources. In addition, it facilitates moving between test and production data sources.

To specify connection information for a data source:

Procedure

1. You can open the **Insert Connection** window in one of the following ways:
 - From the **Project Explorer** view, right-click the **Connections** node. Select **Insert Connection** from the pop-up menu. The **Insert Connection** window opens.
 - From the **Project Explorer** view, double-click the **Connections** node. The **Insert Connection** window opens.
2. From the list of available data sources, select the data source that will be associated with this connection information alias.
3. Specify a unique name for this connection information in the **Connection name** field.
4. Click **Finish**. The **Project Explorer** view lists the new connection information alias under the **Connections** node in the tree.

Note: You can also set up a connection information alias for a data source by using one of the following methods:

- Drag a data source from the **Workspaces** view to the **Connections** folder of the visual project in the **Project Explorer** view.
- Drag a query from the **Workspaces** or **Repository Explorer** view to the **Queries** node of the visual project in the **Project Explorer** view.

A new data source connection is automatically added to the **Connections** folder of the **Project Explorer** view.

Results

Note: The connection information alias is added with the name ConnectionN. You can edit the name of the connection information alias by right-clicking on the entry and selecting **Rename**.

Working with scenes

A visual dashboard consists of one or more scenes that will display data to users.

About this task

Upon creating a new visual dashboard, you are given one default scene. You can choose to have only one scene in your visual dashboard or you can add additional scenes. Within each scene of your dashboard you can add objects to display data and information to visual dashboard users and you can set up the navigation necessary for users to move from one scene or object to another.

This topic describes the following operations:

- [Adding a scene to a visual dashboard](#)
- [Renaming a scene in a visual dashboard](#)
- [Deleting a scene from a visual dashboard](#)

The following steps outline the tasks that you can complete when working with scenes:

Procedure

1. To add a scene to a dashboard, from the **Project Explorer** view, use the **New Scene** wizard.

You can open the New Scene wizard in one of the following ways:

 - Right-click the **Scenes** folder for the visual dashboard to which you want to add the scene and select **New Scene** from the pop-up menu.
 - Double-click the **Scenes** folder for the visual dashboard to which you want to add the scene.

The **New Scene** wizard opens.

 - a) Specify a unique name for this scene in the **Scene Name** field.

b) Click **Finish**.

The **Project Explorer** view lists the new scene in the visual dashboard project. A new scene canvas opens in the editor.

2. To rename a scene, perform the following steps:

a) Select the scene that you want to rename using one of the following methods:

- From the **Project Explorer** view, right-click the scene that you want to rename and select **Rename** from the pop-up menu.
- With an active scene in the editor window, select **Edit > Scene > Rename**

b) Type the new name for the scene in the highlighted edit box. The scene is renamed and the new name is displayed.

c) Click outside the edit box in the **Project Explorer** view.

The scene is renamed and the new name displays.

Note: Any references to the scene by its original name are not updated.

3. To delete a scene, complete either of the following steps:

Note: You cannot delete the scene if it is the last scene remaining.

a) From the **Project Explorer** view, right-click the scene that you want to delete and select **Delete** from the pop-up menu.

b) With an active scene in the editor window, select **Edit > Scene > Delete**.

The scene is deleted from the visual dashboard project.

Note: Any references in the dashboard to the deleted scene are not updated.

Using scene parameters

Using parameters, you can alter the value of an object's properties in the scene. Scene parameters are available for use by only the scene in which it is defined.

About this task

A scene parameter is available for use by only the scene in which it is defined. Each scene that you create has a **Locals** folder that is added to the **Project Explorer** and includes all the parameters that are defined for the scene.

To create a scene parameter:

Procedure

1. From the **Project Explorer** view, right-click the scene for which you want to create the parameter.

Select **New > Parameter**. The **Insert Parameter** window opens.

2. Type a unique name for the scene parameter in the **Name** field.

3. Select a data type from the **Data type** list box. The data type will describe what type of data will be included in the parameter. You have the following options:

- **Boolean:** True or False
- **Color:** Color
- **DateTime:** Date and time value
- **DateTimeSet:** Multiple data and time values
- **FilePath:** Path to a file
- **FontName:** Valid font type
- **Integer:** Whole number
- **Literal:** As is data. This data type only applies to query parameters

- **Number:** Floating point number
- **NumberSet:** Multiple floating point numbers
- **Percentage:** Value that will be used as a percentage
- **Point:** Single x,y point value
- **PointSet:** Multiple x,y point values
- **PolySet:** Multiple point variables that can be used to construct a polygon
- **Text:** A text string
- **TextSet:** Multiple text strings

The default data type is Integer.

4. If you want to add a description, enter the text in the **Description** field.
5. If you want to specify a default value, select the **Has default value** check box. A field is displayed where you can type a default value for the parameter. This value will be used if no other value is specified.
6. Click **Finish**.

The **Insert Parameter** window closes. The specified parameter is added to the parameters node in the **Project Explorer** view.

Creating scene views

In visual dashboards you can create scene views. Use them to arrange and modify scene objects to adjust the view of the scene to different screen resolutions.

Before you begin

You create a scene on the **Canvas** tab and define the behavior of the scene objects. This view is the default view for the scene.


About this task



You can use scene views to customize the view of a visual project, and you cannot use them to change events and event actions that are defined in your visual project.

For each scene in a visual project you can create any number of required scene views.

To create a scene view:

Procedure

1. Open a scene for which you want to create a scene view on the **Design** tab.
The default view is displayed on the **Default View** tab.
2. In the lower right corner of the **Design** tab, click  to open the **Create Scene View** wizard.
3. From the **Category** list, select the type of the screen, and from the **Templates** list, select the screen resolution for that you want to create a scene view. You can select the preset resolution and go to step 10. To create a custom template, go to step 4.
Tip: To display a scene correctly on different mobile devices, you do not need to create scene views for all possible screen resolutions. You can select one template, for example, **iPhone 5, 5C, 5S** from the **Phone** category and create the landscape and portrait scene views for it. At runtime, the appropriate scene view is selected and scaled to screen resolutions of different mobile devices. To enable scene views scaling, ensure that the **ScaleToWindow** scene property is set to true.
4. To add a specific scene view template, select **Custom** from the **Category** list and click **Add custom scene view template**.
5. Enter the template description in the **View description** field.
6. In the **Width** and **Height** fields, specify the width and height in pixels for your template.

7. Click **OK** to add it to the **Templates** list.
8. If you want to remove a custom scene view, select it and click **Delete custom scene view template**.
9. Select the custom view that you created from the **Templates** list.
10. Click **Finish** to close the **Create Scene View** wizard and open the scene view tab. The scene view tab displays a canvas with a frame that defines the screen area.
11. On the scene view tab, modify the content of the scene to adjust it to the defined screen area. You can modify scenes in the following way:
 - Display or hide the scene objects for the view. To do this, select or clear the corresponding check boxes in the **Visibility** view.
 - Rearrange the position of the scene objects.
 - Modify scene objects properties.All modified properties are marked in bold.
12. When you completed creating the scene view, save it with your visual project.
13. If you want to edit the type or size of the current scene view in your visual dashboard, in the lower right corner of the Design tab, click .
14. If you want to remove the current scene view from your visual dashboard, in the lower right corner of the Design tab, click .

Results

Tip: If you want to display a scene view in the embedded scene container, create scene view of the same size as the embedded scene container and specify its name in the SceneView property of embedded scene and use the AutoZoom property of the embedded scene to zoom the content of the embedded scene automatically when a scene view is applied. For more information about the properties see, the *Embedded Scene* topic.

Scene views selection for different devices

The topic contains useful information about how the appropriate scene view is selected when you run a visual dashboard in a web browser of a mobile device or user machine.

When you run a visual dashboard:

1. QMF receives information about the screen size of a web browser that is launched on a mobile device or user machine.
2. From all scene views that are created for the appropriate scene, the scene views with the orientation that corresponds to the orientation of a web browser are selected.
3. From the list of scene views with the appropriate orientation, the scene view with the size that is close to the size of a web browser is selected.
4. The selected scene view is proportionally scaled by width to the size of a web browser.

Note:

- Scaling is available only in web browsers of mobile devices and only if you use web links to open visual dashboards.
- The default scene view is not scaled because scaling requires the exact size of a scene view, but the exact size of the default scene view is not defined until it is built in runtime.
- The scene views with too small size are not scaled properly. For different mobile devices, the minimum size of a scene view is different. It is recommended to check the scene view in a web browser of a mobile device to ensure its correct displaying and scaling.

Adding navigation features to your visual data objects

You design how the user will move through the dashboard by assigning navigation to scenes and objects.

About this task

Depending on how you want to present your data to your user, you can choose to apply any of the following navigation features to your scenes and objects:

Using the Drilldown wizard

Although it can be done manually, the **Drilldown** wizard automates the process of creating the click event, identifying the destination scene, and mapping the required scene parameters.

About this task

The ability to drill down to additional information is a very important navigation feature. It enables users to click an object and jump to another scene that contains more information that relates specifically to the original object's value. The **Drilldown** wizard is available for all layout objects that can display query result data (support data templates).

Note: You can also create click events, identify the destination scene and map the required scene parameters manually.

To set up a click event to a new destination scene using the **Drilldown** wizard:

Procedure

1. From the **Project Explorer** view, expand the layout object's folder to which you want to add the click event navigation feature. Expand the data template folder for that layout object. Expand the level folder. Right-click the object to which you want to apply the click event and select **Drilldown** from the pop-up menu. You can also open the object in the editor and right-click the object in the editor window. The **Drilldown** wizard opens.
2. The first page of the wizard summarizes the steps necessary to create the click event.
3. Click **Next**.

On the first page of the Drilldown wizard, select the query result data columns whose values you would like to pass through to your target scene. Any data values that you pass can be used by the target scene to deliver additional information that applies specifically to that originating data value. Use the arrow keys to move the data columns from the Available fields list box to the Fields to be displayed list box. Each data column that you add to the Fields to be displayed list is passed to the destination scene as a parameter.

4. Click **Next**

On the second page of the Drilldown wizard, specify the scene to jump to. Choose from the Location radio group to determine whether the drilldown action will jump to a scene in the currently selected dashboard, or in another visual project.

- Select **This dashboard** to jump to a scene in the currently selected dashboard.
- Select **Another visual project** to jump to a scene or a page in another visual project. The **Visual project** field becomes available.

Use the **ellipsis (...)** to browse for the visual project you would like to jump to.

- Click **Retrieve Parameters** to select from the available parameters of the specified visual project.

Note: To open the visual project in a new tab, select the **Open in new tab** check box.

5. Select the destination scene that you want the user to jump to from the **Scene** list box. If the scene does not exist, you will be asked to create one. Specify the viewpoint in the scene by selecting a viewpoint from the **Viewpoint** list box.
6. Click **Next**

On the third page of the Drilldown wizard, specify the destination scene's parameter values. All the parameters that have been defined for the destination scene are listed in the Parameter field. These include parameters that apply only to the destination scene as well as the parameters for all the data columns that will be passed to the destination scene.

7. Click **Finish**.

The click event navigation feature has been set for the layout object.

Inserting viewpoints

Insert viewpoints in the navigation to isolate a particular area of a scene at a particular zoom level. Viewpoints define what part of a scene a viewer sees, and at what magnification.

About this task

You can use viewpoints as targets in jumps between scenes and associate them with a particular zoom level. For example, a scene might present a dashboard map indicating factory locations. You can insert and define viewpoints for each region of the dashboard, thereby allowing users to move quickly to the location and magnification for the region they are interested in.

In addition to providing a navigation tool, named viewpoints can be the target of an event action. For example, you can specify the display of a particular viewpoint if a user double-clicks an object.

To insert a viewpoint, perform the following steps:

Procedure

1. Open the **New Viewpoint** window in one of the following ways:

- In the **Canvas** mode in the visual dashboard editor, select **Insert > Insert Viewpoint** to open the **New Viewpoint** window.
- From the **Project Explorer** view, right-click on the **Locals** node for the scene to which you want to add a viewpoint and select **Insert Viewpoint** from the pop-up menu.
- From the **Project Explorer** view, right-click on the scene for which you want to create the viewpoint and select **New > Viewpoint**

The **New Viewpoint** window opens.

2. Type a name for the viewpoint in the **Name** field.

3. Type the location of the x-coordinate in the **X location** field.

The X-coordinate specifies the distance from the center of the scene along the horizontal axis. The default value is 0.

4. Type the value of the y-coordinate in the **Y location** field.

The y-location is the number of inches from the center of the display.

The Y-coordinate specifies the distance from the center of the scene along the vertical axis. The default value is 0.

Note: The default unit of measurement used for the X and Y coordinates depend on the regional standards specified for the client machine. You can change the unit of measurement by typing its abbreviation after the value. For more information about units of measure, see the reference documentation in the product help.

5. Specify the zoom level of the viewpoint in the **Zoom level** list.

The default value is 100%.

6. Click **Finish**.

The **New Viewpoint** window closes.

Results

To modify a viewpoint:

1. In the **Project Explorer** view, locate the viewpoint that you want to modify and double-click it. The **Edit Viewpoint Properties** window opens.
2. Type changes in the **X location** and **Y location** fields.
3. Specify a new zoom level in the **Zoom level** list.
4. Click **OK**.

Note:

You can restore the scene to the default viewpoint by pressing the **Home** key.

The default viewpoint (0,0, 100% zoom) is a helpful orientation point when you are developing a scene. In a large scene, when you want to return to the center, you can press the **Home** key to take you back to the default viewpoint.

You can set a viewpoint for a scene from the **Default Viewpoint** folder in the **Scenes** folder. Right-click **Default Viewpoint** and select **Properties** to change the default viewpoint properties or select **Insert Viewpoint** to create a viewpoint.

Modifying viewpoints

You can modify the viewpoints.

About this task

To modify a viewpoint:

Procedure

1. Open the **Edit Viewpoint Properties** window in one of the following ways:
 - In the **Project Explorer** view, locate the viewpoint that you want to modify and double-click it.
 - In the **Project Explorer** view, right-click the viewpoint that you want to modify and select **Edit Viewpoint**.

The **Edit Viewpoint Properties** window opens.

2. Type changes in the **X location** and **Y location** text boxes.
3. Specify a new zoom level in the **Zoom level** list.
4. Click **OK**.

The Edit Viewpoint Properties window closes.

The default viewpoint (0,0, 100% zoom) is a helpful orientation point when you are developing a scene.

To restore the scene to the default viewpoint:

- a. Right-click in the editor view and select **Jump to....** The Jump to window opens.
- b. Select the default and click **OK**.
- c. From the **Project Explorer** view, right-click the default viewpoint and select **Jump to Viewpoint** from the pop-up menu.
- d. Press the **Home** key.

Results

You have modified the viewpoint.

Using embedded scene objects

An embedded scene object is a container through which you can view another scene or a part of it and then navigate to that scene.

About this task

The scene where you place the embedded scene object is considered the source scene. The destination scene is the embedded scene. When you insert an embedded scene object it appears in the **Scenes** folder in the **Project Explorer** view.

An embedded scene differs from a jump or a hyperlink in the following ways:

- You see the embedded scene from the source scene.
- You can pass parameter information from the source scene to the embedded scene.

You can use parameters to pass information from one scene to another scene using the embedded scene object. For example, if a source scene displays a scatter chart of automobile sales, you can define each point on the chart as embedded scene object that contains a destination scene displaying sales data for a particular manufacturer. To create this example, you pass the manufacturer name as a parameter to the embedded scene using the embedded scene object.

Results

Embedded scene objects have several properties that you might want to modify, including:

- **ZoomPercentage.** Sets the zoom percentage of the embedded scene as viewed through the embedded scene object. You might want to reduce the zoom so that you see more of the embedded scene.
- **SceneCenter.** Sets the position of the embedded scene object over the embedded scene. By default, this property is set to the center of the embedded scene. You can change this value to display another area of the embedded scene. For example, you might have several charts on one scene, but want to display only one of them through the embedded scene object.
- **BorderStyle.** Sets the style of the border around the embedded scene object. You can choose a border style, or choose no border.

Inserting a simple embedded scene object

A simple embedded scene connects two scenes.

Before you begin

For example, a source scene might display an overview of five mutual funds, with small charts for each fund. Each of the five charts can be an embedded scene that when selected will show a full-size display of a fund.

About this task

To insert a simple embedded scene object in your visual dashboard scene:

Procedure

1. From the **Project Explorer**, open the source scene in the **Canvas** mode.
2. From the **Palette** view, click the **Embedded Scene** object.
3. Click in the editor window to insert the embedded scene object.

The **Embedded Scene Wizard** opens.

4. Specify whether you want to use a scene from the current Visual Project or from a previously saved one.

If you want to open a scene from another Visual Project select the corresponding check box and select a project.

If you want to use a scene from the current project make sure that the **Another Visual Project** check box is cleared.

5. Select the name of the scene that you will embed from the **Embedded scene** list box.
6. Select the **Navigate to embedded scene when clicked with the mouse** check box to automatically generate an event action for the embedded scene object. The generated event action specifies that a mouse click the embedded scene object will take a user to the embedded scene. The **Navigate to embedded scene when clicked with the mouse** check box should be cleared if you want to specify a different event action for the embedded scene object.
7. Specify the location and zoom level of the embedded scene in the **X location**, **Y location**, and **Zoom** fields.
8. Click **Finish**.

The **Embedded Scene Wizard** closes and the new embedded scene is displayed in the editor window.

Inserting a parameterized embedded scene object

A parameterized embedded scene object connects two scenes and passes information from one scene to another.

About this task

For example, if a source scene displays a scatter chart of automobile sales, you can define each point on the chart as an embedded scene object that contains a destination scene displaying sales data for a particular manufacturer. You pass the manufacturer name as a parameter to the destination scene using the embedded scene object.

Note: You can use a scene from another Visual Project. If you want to open a scene from another Visual Project check the corresponding check box, click the ellipsis (...) and browse for the visual project you want to use. When you close the Open wizard, path to the selected Visual Project is displayed in the **Another visual project** field.

To create a parameterized embedded scene object:

Procedure

1. Select the scene that you want to embed in the embedded scene object. Choose **Insert > Scene** to create a new scene or open an existing scene by double clicking it in the **Project Explorer** view. If this is a new scene, or you have to edit the scene, add the text and graphical elements.
2. Specify that the embedded scene will receive parameters. Expand the folder for the scene that you want to embed. Expand the **Locals** folder for the scene. Right-click the **Parameters** folder and select **Insert Parameter**. The **Insert Parameter** window opens.
3. Enter the name of the parameter in the **Name** field and select a data type for the parameter from the **Data type** list box. You can enter a description for the parameter in the **Description** field. Select the **Has default value** if you would like to specify a default value for the parameter. A field where you will specify the default value opens. Click the ellipsis (...) for help in specifying the default value. The button that is provided depends on the data type that is selected. Click **Finish**.

The **Insert Parameter** window closes.

4. Select the source scene where you will place the embedded scene object. Select **Insert > Scene** to create a new scene or open an existing scene by double clicking it in the **Project Explorer** view. The source scene is displayed in the editor.
5. Add an embedded scene object to the source scene. From the **Containers** palette in the **Palette** view, double-click **Embedded Scene**.
The **Embedded Scene Wizard** opens.
6. Select the name of the scene that you will embed from the **Embedded scene** list box.
7. Select the **Navigate to embedded scene when clicked with the mouse** check box to automatically generate an event action for the embedded scene object. The generated event action specifies that a mouse click the embedded scene object will take a user to the embedded scene. The **Navigate to**

embedded scene when clicked with the mouse check box should be cleared if you want to specify a different event action for the embedded scene object.

8. Specify the location and zoom level of the embedded scene in the **X location**, **Y location**, and **Zoom** fields.
9. Click **Finish**.
The **Embedded Scene Wizard** closes and the new embedded scene is displayed in the editor window.
10. Select or create an object that can receive user input, for example a combo or list box.
11. Set the parameter value that will be passed to the embedded scene. In the **Project Explorer** view, expand one or more source scene folders to expose the embedded scene object. Expand the embedded scene object's folder. You will see the name of the embedded scene. Click the name of the embedded scene. The parameters that you have specified for the embedded scene are listed in the **Properties** view. Any default values are also listed. You can specify the values in the **Properties** view. Double-click the **Value** field to open the **Expression Designer**. The **Expression Designer** will help you specify the value for the parameter. If you used a list to accept user input then you must select a property value of the list to include in the parameter value.

Setting up event action navigation

By assigning an event to a scene object or to any graphic object that can be inserted in a scene, you set up the object to be responsive to a user's actions as they navigate the dashboard.

Events are actions that are made by the user of the dashboard. Events include user actions such as a mouse click, mouse movements, or the use of keyboard keys. At the same time that you assign an event to an object, you also specify additional actions that will be triggered when the event occurs. Additional actions that you can define for an event include jumping to a new scene, setting a parameter value or issuing a message. As a user navigates the dashboard, when an event that you have assigned to an object occurs (such as a click on a button object), the additional actions that you have defined for the event are triggered.

The following tasks describe assigning events to objects and specifying the actions that will be triggered when the event occurs:

Assigning events and setting up event actions

You use the **Events** view to assign an event to an object and set up the event actions.

About this task

The **Events** view displays the events tree with the events that can be assigned to the objects of a visual project and the list of event actions that are defined for the events.

You define event actions to objects to provide scene interactivity and navigation. As a user navigates your dashboard, mouse actions trigger predefined events, such as jumps between scenes, executing of SQL statements, or the launching of other applications.

To assign an event to an object and set up event actions:

Procedure

1. In most cases, the **Events** view opens with the **Visual Designer** perspective. If the **Events** view is not open, select **Window > Show Views > Events** to open the **Events** view.
2. Select an object in your visual project (either in the scene editor or in the **Project Explorer** view). The events that can be assigned to the selected object are displayed in the events tree in the upper pane of the **Events** view.
3. You can add an event to the selected object in one of the following ways:
 - In the events tree, double-click the event that you want to assign to the object. The **[objectname] Behavior** window opens.

- In the events tree, select the event that you want to assign to the object and click the **Add New Action** button from the lower pane of the **Events** view. The **Add New Action** window opens. Then go to step 5 of the procedure.

Note: To quickly find the required event in the events tree, you can type the name of the event in the filter text field.

4. The event that will be assigned to the object is listed in the **Event** list box. You can select a different event from the drop-down list of available events. The event actions that were defined for the event are listed in the **Actions** list. To add an action, click the **Add** icon. The **Add New Action** window opens. The actions that can be triggered by the event are displayed in the actions tree.
5. In the actions tree, select an action that you want to add to the event. You are prompted for additional information depending on the action that you choose.

Note: To display only event actions that are available for the selected event, ensure that the **Hide disabled actions** check box is selected.

To quickly find the required action in the actions tree, enter the action name in the filter text field.

6. The event action that you define is listed in the **Action** cell of the list box. For each action you can also specify condition and description.

The condition that you specify must be met before the action will occur. To specify a condition, click the **Condition** cell of the list box. Type an expression or click the ellipsis (...) to open the **Expression Designer**. You can use the **Expression Designer** to help you create the conditional expression.

Specify the description so that users can easily determine the purpose for which the action is applied.

7. You can specify that an event will trigger a series of multiple actions. Repeat the add process for each action that you want to add. Each action that you define is listed in the **Actions** list box. The actions that you define for an event are performed in the same order as they are listed in the **Actions** list box. You can use the **Insert after selected item** and **Insert before selected item** buttons to position new actions in the list. You can use the **Move Action Up in List** and **Move Action Down in List** icons in the toolbar to move the actions in the list to new positions. You can use the **Copy Action** and **Paste Action** icons in the toolbar to create duplicate actions.
8. You can delete any action that has been defined for the event. Select one or more actions from the **Actions** list box. Click the **Remove Selected Actions from List** icon. The action is deleted.
9. You can view or modify the details of an action. Select the action from the **Actions** list box. Click the **Edit Selected Action** icon. Depending on the action that you selected, a unique window opens that shows the details of the action. You can make changes to the details and save the changes.
10. Click **OK**. The **[objectname] Behavior window** closes.

Results

The event to which you have added one or more actions is shown in bold font in the events tree in the upper pane of the **Events** view. The list of actions added to the event is displayed in the lower pane of the **Events** view.

You can assign more than one event to an object. To assign another event to the object, select the event and repeat this process.

Assigning jump events to objects in your visual dashboards

Use the **Jump to new location** event action to assign and define *jump events* to objects in your visual projects. At runtime, jump events take the user from an object in the dashboard or application to another scene or viewpoint of a scene or to a visual report.

Before you begin

You must create both the source and the destination dashboard scenes or viewpoints before creating the jump event between them. Additionally, if the jump event you are defining takes the user to a Visual Report, you must create the report before creating the jump event to that report.

About this task

Jumps can carry context information to the new location. For example, you can specify a scene parameter to be set at jump time and base the parameter on a calculated value or category, such as store location. Thus, a destination scene showing sales revenues can be based on store location and display different data depending on what context the user is coming from.

Note: You must create both the source and the destination project scenes or viewpoints before you can create a jump between them. You must also create a visual report before you can create a jump to the visual report.

To assign and define a jump event for an object in your visual project, perform the following steps:

Procedure

1. Select the object in the editor window and double-click the event that you want to trigger the jump from the **Events** view.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** window opens.
3. Select **Jump to new location** in the actions tree.
Note: To quickly find the required event action, type the name of the action in the filter text field.
4. Click **Next**.
The Jump to new location action parameters page opens.
5. Specify whether the destination of the jump will be within the current visual project or to another. Select **This visual project** to specify that the destination for the jump will be to a scene or viewpoint within the current project. Select **Another visual project** to specify that the destination for the jump will be another visual project.
6. If you selected **This visual project**, the **Scene** and **Viewpoint** fields become available. From the **Scene** list box, select the destination scene name. From the **Viewpoint** list box, select a viewpoint. If there are no viewpoints in the scene, the default is used.
7. If you selected **Another visual project**, the **Visual project** field becomes available. Specify the name of the destination visual project in the **Visual project** field. Click the ellipses button to search for and open a project from a repository or a file.
The **Open** window opens. Navigate to and select the visual project that you want the event action to jump to and click **Finish**. The Open window closes.
8. Optional: If the visual project that you selected has parameters, click the **Retrieve Parameters** button.
Note: The parameters that must be set when the event occurs are listed.
9. Specify a value for each of the parameters in the **Value** cell for each parameter.
10. From the **Open in** area select the required option:
 - a) **Same tab** to open another visual project in the current tab.
 - b) **New tab** to open another visual project in a new tab in QMF for Workstation or in a new browser tab in QMF for WebSphere.
 - c) **New window** to open another visual project in a new tab in QMF for Workstation or in a new browser window in QMF for WebSphere.
11. Click **Finish**. The **Jump to new location** event action is added. The **Add New Action** window closes. Control returns to the **[objectname] Behavior** window.
12. From the **[objectname] Behavior** window you can add conditional expressions to the jump to a new location action. You can define multiple jump to new location actions each with a unique condition. Once a jump to a new location does occur, any subsequent actions defined for the object will not occur.

13. From the **[objectname] Behavior** window, click the **Apply** button to apply all actions that have been added for the event. Click **OK** to close the **[objectname] Behavior** window.

Setting global and scene parameter values

Use the **Set values** action to set global and scene parameter values.

About this task

When an event occurs, you can set global and scene parameters to new values. The new values can be constants, functional expressions of data values, or other parameters (global, scene, or event parameters). You must create the parameters that will be used.

To set parameter values when an event occurs:

Procedure

1. Select the object in the editor window and double-click the event that you want to trigger the jump from the **Events** view.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** window opens.
3. Select **Set values** in the actions tree.
Note: To quickly find the required event action, type the name of the action in the filter text field.
4. Click **Next**.
The **Set values action parameters** page opens.
5. Click in the **Parameter** field and select a parameter name from the list of available parameters. All the parameters that you have created and are available to the scene are listed.
6. Click in the corresponding **Value** field and select a value from the list box. The parameter value must match the parameter type.
7. To add a parameter, click the **Add New Entry** button. Type the name of the new parameter in the **Parameter** field. Specify the value for the parameter in the **Value** field.
8. To delete a parameter, select the parameter and click the **Delete** icon.
9. Click **Finish**. The **Set values** action is added. The **Add New Action** window closes.
10. From the **[objectname] Behavior** window, click the **Apply** button to apply all actions that have been added for the event.
11. Click **OK** to close the **[objectname] Behavior** window.

ExternalEvent events

ExternalEvent events occur when a user sends some event to an embedded scene object by using the **Send to Embedded Scene** event action.

You can assign ExternalEvent events to the scene objects.

You can incorporate ExternalEvent events in scene objects to trigger different event actions. For more information about event actions, see [“Assigning events and setting up event actions” on page 363](#)

Adding a Navigate to embedded scene event action

Use the **Navigate to embedded scene** event action to move the user to a scene or viewpoint of a scene that has been embedded in an **Embedded Scene** object.

About this task

Embedded scenes are added and set up automatically when a user clicks directly the embedded scene object. This is the default behavior.

You might want to add a **Navigate to embedded scene** event action manually for the following scenarios:

- To specify an event (such as a click) on a different object (such as a button) to trigger entering the embedded scene of an **Embedded Scene** object.

For example, you can insert a **Button** object in a dashboard scene and specify that a specific embedded scene (that has been inserted in the same dashboard scene) will be entered when a user clicks the button.

- To add an event (in addition to the **Click** event) to trigger entering the embedded scene of an **Embedded Scene** object.
- To specify an event (other than the **Click** event) to trigger entering the embedded scene of an **Embedded Scene** object.

For example, you might want a **DbClick** event to trigger entering the embedded scene of an **Embedded Scene** object.

To add and define a **Navigate to embedded scene** event action to an object:

Procedure

1. Select the object in the editor window. Double-click the event that you want to trigger the action from the **Events** view. The **[objectname] Behavior** window opens.
2. Click the **Add** icon. The **Add New Action** window opens.
3. Select **Navigate to embedded scene** in the actions tree.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. Click **Next**. A window requesting the embedded scene action parameters opens.
5. Select an embedded scene object from the **Embedded scene** list of available embedded scene objects. Each embedded scene object that you have inserted in the current source scene is listed.
6. Select a viewpoint from the **Viewpoint on [scenename]** list of available viewpoints for the embedded scene. Each viewpoint object that you have created for the embedded scene is listed. If there are no viewpoints in the scene, the default viewpoint is listed.
7. Click **Finish**. The **Navigate to embedded scene** event action is added. The **Add New Action** window closes.
8. From the **[objectname] Behavior** window, click **Apply** to apply all actions that have been added for the event. Click **OK** to close the **[objectname] Behavior** window.

Adding a Navigate back event action

Use the **Navigate back** event action to return a user from the current object to the previous scene or to the previous viewpoint of a scene. For example, you can specify that when a user performs an event (such as a mouse click) on an object (such as a button), the result is that they are sent back to the previous scene.

About this task

To add a **Navigate back** event action to an object:

Procedure

1. Select the object in the editor window and double-click the event that you want to trigger the jump from the **Events** view.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** window opens.
3. Select **Navigate back** in the actions tree.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. Click **Finish**. There are no additional parameters required for the **Navigate back** action. The **Navigate back** action is added. The **Add New Action** window closes.

5. From the [objectname] Behavior window, click **Apply** to apply all actions that have been added for the event. Click **OK** to close the [objectname] Behavior window.

Adding an Execute SQL statement event action

Use the **Execute SQL statement** action to run an SQL statement after a specified event occurs.

About this task

To add an **Execute SQL statement** event action to an object:

Procedure

1. Select the object in the editor window. The **[objectname] Behavior** window opens.
From the **Events** view, double-click the event that you want to trigger the **Execute SQL statement** action.
2. Click the **Add** icon. The **Add New Action** window opens.
3. Select **Execute SQL statement** in the actions tree and click **Next**.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. Enter the SQL statement in the **SQL statement** field.

The SQL statement can contain parameters. If you want to include global or scene parameters in your SQL statement, you must enter the SQL statement as an expression, much as you would enter an expression into a property value. For example, to run the following statement:

```
Update MyTable Set Shipped = 1
Where CustomerID = [value of g_nCustID global variable]
```

You would enter the following text in the SQL Statement field:

```
"Update MyTable Set Shipped = 1 Where CustomerID = " + g_nCustID
```

5. Optional: If the SQL statement affects the content of a scene viewed by your user, be sure to select the **Refresh the display on completion** check box.
6. Optional: Select **Display a message box indicating the result of the operation** if you want to display a message about the success or failure of the execute SQL statement.

If you decide to display a message box indicating the result of the operation, enter the text of the message that you want to display in both the **Success** field and the **Failure** field.

Note: Use \$ (1) to include the system output message as part of the success or failure message text.

7. Select **Save the success or failure (and any return value) in a specified result parameter** to save the result of the executed SQL statement in either a local or global parameter.

When you select this check box the **Result parameter** lists become available.

The local and global parameters that have been created for your dashboard are listed in the **Result parameter** drop-down lists.

Select the local or global parameter that you want to use to store the result information.

The results will be stored as follows:

- If the SQL statement returns one or more values, the first column in the first row is placed in your selected global or scene parameter. For example, the statement `select count(*) from MyTable` will place the count in your selected parameter. The statement `select A, B, C from MyTable` will place the value of A from the first row in your selected parameter.
- If the statement does not produce results, the success or failure of the statement execution is placed in your selected parameter. For example, the statement `update MyTable Set A = 3 where CustomerID = 1` will place true or 1 in your selected parameter if the statement executed successfully, or false or 0 if the statement failed to execute. When this parameter is initially created, it has a data type of Boolean. You can modify the name and data type of the global parameter.

8. Click **Finish**.

The **Execute SQL statement** action is added. The **Add New Action** window closes.

9. From the **[objectname] Behavior** window, click **Apply** to apply all actions that have been added for the event. Click **OK** to close the **[objectname] Behavior** window.

Results

You have added an *Execute SQL statement* event action to an object.

Adding an Execute procedure command event action

Use the **Execute procedure** action to run a procedure command after the specified event occurs

About this task

To add an **Execute procedure** event action to an object:

Procedure

1. Select the object in the editor window. Double-click the event that you want to trigger the action from the **Events** view. The **[objectname] Behavior** window opens.
2. Click the **Add** icon. The **Add New Action** window opens.
3. Select **Execute procedure** in the actions tree and click **Next** to open the **Execute procedure action parameters** page.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. Use the **From Repository** field to enter the procedure statement name and location.
5. Select the **Hide procedure result windows** check box to hide the results window after running the procedure statement.
6. Select the **Refresh display on completion** check box to refresh the dashboard display after running the procedure statement.
7. Select the **Display a message box indicating the result of the operation** check box to display the success or failure of the run procedure statement.

If you check this check box:

- Type the text that you want displayed upon a successful run in the **Success** field. You can use \$(1) in your message to include the system output message as part of the success message text.
 - Type the text that you want displayed upon a failed run in the **Failure** field. You can use \$(1) in your message text to include the system output message as part of the failure message.
8. Check the **Save the success or failure (and any return value) in a specified result parameter** check box to save the result of the executed procedure statement in either a local or global parameter. When this check box is selected the **Result parameter** lists become available. The local and global parameters that have been created for your dashboard are listed in the **Result parameter** drop-down lists. Select the local or global parameter that you want to use to store the result information.

The results will be stored as follows:

- If the procedure statement returns one or more values, the first column in the first row is placed in your selected global or scene parameter. For example, the statement `select count(*) from MyTable` will place the count in your selected parameter. The statement `select A, B, C from MyTable` will place the value of A from the first row in your selected parameter.
 - If the procedure statement does not produce results, the success or failure of the statement execution is placed in your selected parameter. For example, the statement `update MyTable Set A = 3 where CustomerID = 1` will place true or 1 in your selected parameter if the statement executed successfully, or false or 0 if the statement failed to execute. When this parameter is initially created, it has a data type of Boolean. You can modify the name and data type of the global parameter.
9. Click **Finish**. The **Execute procedure** action is added. The **Add New Action** window closes.

10. From the **[objectname] Behavior** window, click **Apply** to apply all actions that have been added for the event. Click **OK** to close the **[objectname] Behavior** window.

Adding an Execute JavaScript event action

Use the **Execute JavaScript** action to run JavaScript after the specified event occurs. For example, you can run JavaScript to open a new window or start the system print dialog.

About this task

To add an **Execute JavaScript** event action to an object:

Procedure

1. From the **Events view**, select the object in the editor window and double-click the event that you want to trigger the **Execute JavaScript** action.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** dialog opens.
3. Select **Execute JavaScript** in the actions tree and click **Next** to open the **Execute JavaScript action parameters** page.
Note: To quickly find the required event action, type the name of the action in the filter text field.
4. Type JavaScript commands in the **JavaScript** field and click **Finish**.
The JavaScript command action is added. The **Add New Action** window closes.
5. From the **[objectname] Behavior** window, click **Apply** to apply all actions that have been added for the event.
6. Click **OK** to close the **[objectname] Behavior** window.

Results

You have added an Execute JavaScript event action to an object.

Adding a Print scene event action

Use the **Print scene** to add a print event action. For example, you can dictate that when the user clicks a **Button** object on a particular scene, the result will be to print the values that are displayed.

About this task

To incorporate a **Print scene** event action in an object:

Procedure

1. From the **Events view**, select the object in the editor window and double-click the event that you want to trigger the **Print scene** action.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** dialog opens.
3. Select **Print scene** in the actions tree.
Note: To quickly find the required event action, type the name of the action in the filter text field.
4. Click **Next**.
The **Print scene action parameters** page opens.
5. Click **Finish**.
The **Print scene** action is added and the **Add New Action** window closes.

6. From the **[objectname] Behavior** window, click **Apply** to apply all actions that have been added for the event.
7. Click **OK** to close the **[objectname] Behavior** window.

Adding a Send Email event action

Use the **Send Email** event action to send a scene to an email recipient after the assigned event occurs. For example, you might incorporate this action when the user clicks **Preview** on a particular scene, so as a result the scene is sent to an email recipient.

About this task

To add a **Send email** event action to an object:

Procedure

1. From the **Events** view, select the object in the editor window. Double-click the event that you want to trigger.
2. In the **[objectname] Behavior** window, click the **Add** icon. The **Add New Action** window opens.
3. Select **Send Email** in the actions tree.
Note: To quickly find the required event action, type the name of the action in the filter text field.
4. Click **Next**. A dialog requesting the **Send Email** parameters opens.
5. Specify the address that the email will be sent to in the **To** field.
6. Specify the address that a carbon copy of the email will be sent to in the **Cc** field.
7. Specify the subject of the email in the **Subject** field.
8. Enter the body text of the email in the **Message** field.
9. Specify the address that will send the email in the **From** field.
10. To specify SMTP server parameters, click **SMTP settings**. In the SMTP Settings window, specify whether to use a server predefined by administrator or a custom one.
 - a) If you decide to use a predefined SMTP server, select the required server from the list of predefined servers or click **Add Server** to create one.
 - b) If you decide to use a custom server, specify server parameters including its address and port number. If the SMTP server requires authentication, select **Use SMTP authentication**, and then specify user name and password. If it is necessary to specify additional access preferences, click **Advanced** to open the **Advanced SMTP Settings** window where you can add required keywords supported by javamail. Click **OK**.Click **OK** to close the **SMTP Settings** window.
11. To include an attachment with the email, create a local or global parameter that specifies the file path and file name of the item that you want to send.
12. Select whether the file path is taken from a global or local parameter and what parameter is used in the **Take file path from parameter** drop-down lists.
Note: Valid parameter types are **FilePath**, **Literal**, and **Text**.
13. Click **Finish**. The **Send Email** action is added. The **Add New Action** window closes.
14. From the **[objectname] Behavior** window, click the **Apply** button to apply all actions that have been added for the event.
15. Click **OK** to close the **[objectname] Behavior** window.

Adding a Send event to embedded scene action

Use the **Send Event to Embedded Scene** action to initiate a specific event in an embedded scene

About this task

The **Send Event to Embedded Scene** action generates an event with a specified ID and sends it to the specified embedded scene.

To add a **Send event to embedded scene** event action to an object:

Procedure

1. Select the object in the editor window. Double-click the event that you want to trigger the action from the **Events** view. The **[objectname] Behavior** window opens. Click the **Add** icon. The **Add New Action** window opens.
2. Select **Send Event to Embedded Scene** in the actions tree.
Note: To quickly find the required event action, type the name of the action in the filter text field.
3. Click **Next**. A dialog requesting the **Send Event to Embedded Scene** action parameters opens.
4. Specify at least one embedded scene in the **Target scenes** field.
5. Optional: In the **Result options** section, specify the global parameter to be used as an external event identifier in the target scene.

You can add a new parameter by clicking the **Add New Parameter**.

Note: The **ExternalEventID** global parameter is used by default.

6. Select the parameter that you want to use from the **Result parameter** drop down lists.
7. Specify a unique event identifier in the **Event ID** field and click **Finish**. The **Add New Action** window closes.
8. From the **[objectname] Behavior** window, click the **Apply** button to apply all actions that have been added for the event.
9. Click **OK** to close the **[objectname] Behavior** window.
10. From the **Project Explorer** view, select a scene connected with the target embedded scene.
11. From the **Events** view select the **ExternalEvent** item and double-click it.

In the **[objectname] Behavior** window, set up the event actions.

Consider the example:

- You have the scenes: **Scene1** and **Scene2** and the buttons: **Button1** and **Button2** on **Scene1**.
- Add the following actions to the buttons: **Button1** action is **ExportToPDF** and **Button2** action is **SendEmail**.
- Set the values of **Event ID**, type any string there, for example, **Button1 Event ID** is *Btn1ExportToPDF* and **Button2 Event ID** is *Btn2SendEmail*.
- Add different **ExternalEvent** actions to the **Scene2** and set the following conditions of the **ExternalEventID** parameters:

| Action | Condition |
|-------------|-------------------------------------|
| ExportToPDF | =ExternalEventID ='Btn1ExportToPDF' |
| SendEmail | =ExternalEventID=' Btn2SendEmail ' |

These actions will be applied to the content of the **Scene2**.

Click **OK** to close the **[objectname] Behavior** window.

Adding an Execute shell command event action

Use the **Execute shell command** action to run a shell command after a specified event occurs. The **Execute shell command** event action allows you to run a shell command like that performed using the operating system Run command from the Start menu.

About this task

For example, you can run a shell command to open Notepad or an email program.

To add an **Execute shell command** event action to an object:

Procedure

1. Select the object in the editor window and double-click the event that you want to trigger the jump from the **Events** view.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** window opens.
3. Select **Execute shell command** in the actions tree.
Note: To quickly find the required event action, type the name of the action in the filter text field.
4. Click **Next**.
The **Execute shell command action parameters** page opens.
5. Type the command in the **Command** box or click **Browse** to browse and select a file that contains the commands.
Note: You can use expressions to specify commands. For example, type `="%SystemRoot%\notepad.exe " + Var1` to open a file, the path to which is stored in the `Var1` parameter, in Notepad.
6. Click **Finish**. The **Execute shell command** action is added. The **Add New Action** window closes.
7. From the **[objectname] Behavior** window, click the **Apply** button to apply all actions that have been added for the event.
8. Click **OK** to close the **[objectname] Behavior** window.

Adding a Modify Data event action

Use the **Modify Data** event action to insert, update, or delete data in the database. This event action allows you to modify data without writing SQL statements manually.

About this task

To add a **Modify Data** event action, perform the following steps:

Procedure

1. Select the object in the editor window. In the **Events** view, double-click the event that you want to trigger the **Modify Data** event action.
The **[objectname] Behavior** window opens.
2. Click **Add New Action**.
The **Add New Action** window opens.
3. Select **Modify Data** in the actions tree and click **Next** to open the **Modify Data action parameters** page.
4. From the **Operation Type** area, select the type of operation:
 - **Insert** to specify the data that you want to add as a row.
 - **Update** to specify the data that you want to update under specified conditions.
 - **Delete** to specify the data that you want to delete as a row under specified conditions.
5. From the **Select table** tree, select the table in which you want to modify data:
 - Expand **All Data Sources** to use a data source different from a data source that is displayed in **Connections** in **Project Explorer**. The data source from which you select the table is added to **Connections** automatically.
 - Expand one of the connections that are already displayed in **Connections** to use a data source from an existing connection.
6. If you want the scene updated automatically after the **Modify Data** action is implemented, select the **Refresh display on completion** check box.
7. Click **Next** to set the action parameters depending on the selected operation type:

- If you select **Insert**:
 - a. On the **Value parameters for the Insert operation** page of the wizard, select the columns into which you want to insert data.
 - b. Enter the expressions to specify the values for the columns by using exact data or calculated expressions.
 - c. In the lower part of the page, select **Preview SQL statement with values** if you want to view the SQL statement with specified object properties or results of expressions, or **Preview SQL statement with expressions** if you want to view the SQL statement with entered expressions.
 - If you select **Update**:
 - a. On the **Value parameters for the Update operation** page of the wizard, select the columns in which you want to update data.
 - b. Enter the expressions to specify the values for the columns by using exact data or calculated expressions.
 - c. In the lower part of the page, select **Preview SQL statement with values** if you want to view the SQL statement with specified object properties or results of expressions, or **Preview SQL statement with expressions** if you want to view the SQL statement with entered expressions, and click **Next**.
 - d. On the **Condition parameters for the Update operation** page, click **Add Row Condition** to enter a new condition.
 - e. In the **Expression** column, select an item from the list or enter the expression manually.
 - f. In the **Operation** column, enter an operator or select it from the list.
 - g. In the **Condition** column, specify a value manually or by using the expression designer.
 - h. Enter conditions in one column and separate rows to use the **And** connector, or in separate columns and one row to use the **Or** connector.
 - i. In the lower part of the page, select **Preview SQL statement with values** if you want to view the SQL statement with specified object properties or results of expressions, or **Preview SQL statement with expressions** if you want to view the SQL statement with entered expressions.

Restriction: All conditions in SQL statements are evaluated one by one in direct order. Changing the precedence by grouping conditions is not supported.
 - If you select **Delete**:
 - a. On the **Condition parameters for the Delete operation** page, click **Add Row Condition** to enter a new condition.
 - b. In the **Expression** column, select an item from the list or enter the expression manually.
 - c. In the **Operation** column, enter an operator or select it from the list.
 - d. In the **Condition** column, specify a value manually or by using the expression designer.
 - e. Enter conditions in one column and separate rows to use the **And** connector, or in separate columns and one row to use the **Or** connector.
 - f. In the lower part of the page, select **Preview SQL statement with values** if you want to view the SQL statement with specified object properties or results of expressions, or **Preview SQL statement with expressions** if you want to view the SQL statement with entered expressions.

Restriction: All conditions in SQL statements are evaluated one by one in direct order. Changing the precedence by grouping conditions is not supported.
8. Click **Finish** to close the **Add New Action** wizard.
 9. Click **OK** to close the **[objectname] Behavior** window.

Example

You work with table PRODUCTIONBYREGION (rsbi:/Data Sources/Sample Data Source/Database/Tables/DEMO/PRODUCTIONBYREGION). You want to add an event action that deletes a row if a value in the

PRODUCT column is "PLYWOOD" and a value in the **QTY** column is less than 600 or if a value in the **QTY** column is more than 1000. Specify the condition parameters for the Delete operation as shown in the table below.

| Expression | Operation | Condition | or | Condition |
|------------|-----------|-----------|----|-----------|
| PRODUCT | = | PLYWOOD | = | |
| QTY | | | > | 1000 |
| QTY | < | 600 | = | |

```
DELETE FROM DEMO.PRODUCTIONBYREGION WHERE (PRODUCT = 'PLYWOOD')
AND (QTY <600) OR (QTY >1000)
```

Adding a Show message event action

Use the **Show message** event action to display a message, save the user response, and use it in other actions.

About this task

To add a **Show message** event action to an object:

Procedure

1. Select the object in the editor window and double-click the event that you want to trigger the jump from the **Events** view.
The **[objectname] Behavior** dialog opens.
2. Click the **Add New Action** icon.
The **Add New Action** dialog opens.
3. Select **Show message** in the actions tree and click **Next**.
Note: To quickly find the required event action, type the name of the action in the filter text field.
The **Show message action parameters** page opens.
4. Type the text of the message that you want to issue in **The text of message** field.
5. Specify where the message is issued by using the **Display type** radio buttons. Select **Message box** to display the message in a standard message box. Select **Output view** to display the message in the **Output** view of QMF for Workstation.
6. Select the severity level of the message from the **Logging level** list.
7. If you want to specify extra message box parameters, click **Next**, otherwise go to the step 10.
8. In the **Message Box** area, specify the following parameters:
 - a) In the **Title** field, enter the title that you want to use for the message box. Double-click the field to open the **Expression Designer**.
 - b) From the **Buttons** list, select the button combinations that you want to add to the message box. The indexes of the selected buttons can be saved to parameters.
Tip: The indexes of the buttons begin with 1.
Tip: If a user closes the message box, the index of the last button is received.
 - c) From the **Icon** list, select the icon that you want to be added to the message box.
9. If you want to save the button index of the button that is clicked by a user, perform the following steps:
 - a) To save the index to the existing parameter, in the **Result Options** area, select the **Save result to parameter** check box

- b) From the **Result parameter** list, select the parameter that you want to use.
- c) To add a parameter, click **Add New Parameter** button.
- 10. Click **Finish**. The **Show message** action is added. The **Add New Action** dialog closes.
- 11. From the **[objectname] Behavior** dialog, click **Apply** to apply all actions that was added for the event.
- 12. Click **OK** to close the **[objectname] Behavior** dialog.

Example

If you creates a message with the Yes/No/Cancel combination of buttons, the following indexes can be received:

| <i>Table 34. Buttons and their indexes.</i> | |
|---|----------------|
| Clicked Button | Received Index |
| Yes | 1 |
| No | 2 |
| Cancel | 3 |
| A user closes the message box | 3 |

Adding an Export to Excel event action

Use the **Export to Excel** action to export a specified event to Excel.

Before you begin

See system requirements in the Installing and Managing guide for information about the version of Excel required to support the **Export to Excel** feature.

About this task

To add an **Export to Excel** event action to an object:

Procedure

1. From the **Events** view, select the object in the editor window and double-click the event that you want to trigger.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** window opens.
3. Select **Export to Excel** in the actions tree and click **Next** to open the **Export to Excel action parameters** page.
Note: To quickly find the required event action, type the name of the action in the filter text field.
4. Specify the source component name in the **Data Template** field and click the ellipsis (...).
The **Select data template** window opens.
Select the data source template you want to use and click **OK** to close the window.
The system fills the highest data level and populates the properties and columns entries in the **Export data** text block.
5. Use the **Move Up** and **Move Down** icons in the toolbar to move the entries in the list to new positions.
6. You can delete any entry that has been defined for the export columns. Select an entry from the **Export columns** list box. Click the **Delete Entry** icon. The entry is deleted.
7. In the **Export type** radio group, select one of three radio buttons:
 - **Open**. It opens **Save As** dialog. Enter the name of the file in the **File name** field to save it.

- **Save to a temp file.** It saves the generated file in a temporary directory. The path to this file is recorded in a global or local parameter. You specify the parameter in the bottom of the window.
 - Select either **Global parameters** or **Local parameters** from the **Result parameter** drop-down list.
 - Click **Add New Parameter**. The **Insert Parameter** window opens.
 - Enter a name for the parameter and its description in the **Name** and **Description** fields.
 - Select a parameter type. The valid types are **FilePath**, **Literal**, and **Text**.
 - Select the **Has default value** check box and specify the full file path in the text field; for example, C:\DOCUME~1\<user>\pdf\.
 - **Save to a file.** It automatically saves the generated file in a specified root directory when the event occurs.
8. If you select **Save to a file** check box there are several ways of building an export path, depending on the directory which can be specified in the **Root output directory** field on the **Server-Side File System** page of the **Preferences** window:
 - When the output root directory is specified, the export directory of the file will be combined of the root output directory and the path typed in the **Path** field.
 - In QMF for Workstation when the output root directory is not specified, the export directory of the file will be a combination of the application working directory and the path typed in the **Path** field.
 - In QMF for WebSphere when the output root directory is not specified, the export directory of the file will be a combination of the temporal directory and the file name, the typed path is ignored.
 9. From the **Format type** list, select the Excel file format in which you want to save the exported data. The available options are:
 - **XLS** Microsoft Excel 97-2003 (*.xls)
 - **XLSX** Microsoft Excel 2007 (*.xlsx)
 10. To filter the available dashboard components, specify a string in the **type filter text** field. Only the components that have names that contain the specified string will be displayed in the **Export data** tree. If you selected **Save to a temp file** or **Save to a file**, the **Result options** section is active.
 11. If you did not previously specify the directory parameter, select either **Global parameters** or **Local parameters** from the **Result parameter** drop-down list and click the **Add New Parameter button**. The **Insert Parameter** window opens.
 12. Specify a necessary parameter and click **Finish**. The **Insert Parameter** window closes and control returns to the **Add New Action** window.
 13. Click **Finish**. The **Export to Excel** action is added. The **[objectname] Behavior** window is updated.
 14. Click **OK**. The **[objectname] Behavior** window closes.

Related tasks

[Setting the directory location preference for objects generated by Export actions](#)

You can set preferences in QMF so that objects generated from various QMF actions are exported to a specific location on either the local file system or the server file system.

Adding a Set focus event action

Use the **Set focus** event action to set the focus on any given control name parameter. For example, you might incorporate the **Set focus** event action so that when a user makes a selection, the result is to take the control name as the parameter on which the focus needs to be set.

About this task

To add a **Set focus** event action to an object:

Procedure

1. From the **Events** view, select the object in the editor window and double-click the event that you want to trigger.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** window opens.
3. Select **Set focus** in the actions tree.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. Click **Next**. The **Set focus action parameters** window opens.
5. Specify the control name of the component in one of the following ways:
 - Type the name in the **Component** field;
 - Click **Set from current level** to choose the component.The **Set component dialog** opens. Select the component from the current level and click **OK** to close the Set component dialog.
6. Click **Finish**. The **Set focus** action is added. The **Add New Action** window closes.
7. From the **[objectname] Behavior** window, click the **Apply** button to apply all actions that have been added for the event.
8. Click **OK** to close the **[objectname] Behavior** window.

Results

You have added a *Set focus* event action to an object.

Adding a Reload scene event action

Use the **Reload scene** event action to reload the scene after selecting a translation table language. For example, the user can select a **Translation table** language on a particular scene and have the selected language reflected when the user navigates to another scene.

About this task

To add a **Reload scene** event action to an object:

Procedure

1. Select the object in the editor window and double-click the event that you want to trigger the jump from the **Events** view.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** window opens.
3. Select **Reload scene** in the actions tree.

- Note:** To quickly find the required event action, type the name of the action in the filter text field.
4. Click **Finish**. The **Reload scene** action is added. The **Add New Action** window closes.
 5. From the **[objectname] Behavior** window, click the **Apply** button to apply all actions that have been added for the event.
 6. Click **OK** to close the **[objectname] Behavior** window.

Adding an End session event action

Use the **End Session** event action to end the current session and close the application.

About this task

By adding a **End Session** event action to a visual dashboard, you can close the application from the project without having access to the application's user interface. This is useful when you are accessing a visual dashboard from an embedded window.

To add an **End session** event action to an object:

Procedure

1. Select the object in the editor window and double-click the event that you want to trigger the jump from the **Events** view.

The **[objectname] Behavior** window opens.

2. Click the **Add New Action** icon.

The **Add New Action** window opens.

3. Select **End Session** in the actions tree.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. Click **Finish**. The **End Session** action is added. The **Add New Action** window closes.
5. From the **[objectname] Behavior** window, click the **Apply** button to apply all actions that have been added for the event.
6. Click **OK** to close the **[objectname] Behavior** window.

Adding an Export to PDF event action

Use the **Export to PDF** event action to export specific components of a visual dashboard to a PDF file.

About this task

The **Export to PDF** event action allows you to select specific components of a visual dashboard that you want to export to a PDF file, while excluding those that are unnecessary or extraneous. This is useful for situations where a visual dashboard has a recurring graphical object, like a header, that would take up unneeded space in a printed format.

Restriction: In the HTML5 mode, the content of Google map and map objects cannot be exported in this action.

To add an **Export to PDF** event action to an object:

Procedure

1. Select the object in the editor window and double-click the event that you want to trigger the jump from the **Events** view.

The **[objectname] Behavior** window opens.

2. Click the **Add New Action** icon.

The **Add New Action** window opens.

3. Select **Export to PDF** in the actions tree.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. Click **Next**.

An **Export to PDF** specific window opens.

5. In the **Export type** radio group, select one of three radio buttons:
 - **Open**. It opens **Save As** dialog. Enter the name of the file in the **File name** field to save it.
 - **Save to a temp file**. It saves the generated file in a temporary directory. The path to this file is recorded in a global or local parameter. You specify the parameter in the bottom of the window.
 - Select either **Global parameters** or **Local parameters** from the **Result parameter** drop-down list.
 - Click **Add New Parameter**. The **Insert Parameter** window opens.
 - Enter a name for the parameter and its description in the **Name** and **Description** fields.
 - Select a parameter type. The valid types are **FilePath**, **Literal**, and **Text**.
 - Select the **Has default value** check box and specify the full file path in the text field; for example, C:\DOCUME~1\\LOCALS~1\Temp\\reporter-\pdf\..
 - **Save to a file**. It automatically saves the generated file in a specified root directory when the event occurs.
6. If you select **Save to a file** check box there are several ways of building an export path, depending on the directory which can be specified in the **Root output directory** field on the **Server-Side File System** page of the **Preferences** window:
 - When the output root directory is specified, the export directory of the file will be combined of the root output directory and the path typed in the **Path** field.
 - In QMF for Workstation when the output root directory is not specified, the export directory of the file will be a combination of the application working directory and the path typed in the **Path** field.
 - In QMF for WebSphere when the output root directory is not specified, the export directory of the file will be a combination of the temporal directory and the filename, the typed path is ignored
7. To filter the available dashboard components, specify a string in the **type filter text** field.

Only the components that have names that contain the specified string will be displayed in the **Export data** tree. If you selected **Save to a temp file** or **Save to a file**, the **Result options** section is active.
8. If you did not previously specify the directory parameter, select either **Global parameters** or **Local parameters** from the **Result parameter** drop-down list and click the **Add New Parameter button**.

The **Insert Parameter** window opens.
9. Specify a necessary parameter and click **Finish**.

The **Insert Parameter** window closes and control returns to the **Add New Action** window.
10. Select the parameter that you want to use from the **Result parameter** drop-down lists and click **OK**.

The Add New Action dialog closes.
11. From the **[objectname] Behavior** window, click **Apply** to apply all actions that have been added for the event.
12. Click **OK** to close the **[objectname] Behavior** window.

Related tasks

[Setting the directory location preference for objects generated by Export actions](#)

You can set preferences in QMF so that objects generated from various QMF actions are exported to a specific location on either the local file system or the server file system.

Adding a Create File event action

Use the **Create File** to create a specific content while running a visual project and then use this content for different purposes, for example, sending it in an email.

About this task

To add an **Create File** event action to an object:

Procedure

1. Select the object in the editor window and double-click the event that you want to trigger the jump from the **Events** view.

The **[objectname] Behavior** window opens.

2. Click the **Add New Action** icon.

The **Add New Action** window opens.

3. Select **Create File** in the actions tree and click **Next** to open the **Create File** specific window.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. In the **Output type** radio group, select one of three radio buttons:

- **Open.** It opens **Save As** dialog. Enter the name of the file in the **File name** field to save it.
- **Save to a temp file.** It saves the generated file in a temporary directory. The path to this file is recorded in a global or local parameter. You specify the parameter in the bottom of the window.
 - Select either **Global parameters** or **Local parameters** from the **Result parameter** drop-down list.
 - Click **Add New Parameter**. The **Insert Parameter** window opens.
 - Enter a name for the parameter and its description in the **Name** and **Description** fields.
 - Select a parameter type. The valid types are **FilePath**, **Literal**, and **Text**.
- **Save to a file.** It automatically saves the generated file in a specified root directory when the event occurs.

5. If you select **Save to a file** check box you can build an export path in different ways, depending on the directory which can be specified in the **Root output directory** field on the **Server-Side File System** page of the **Preferences** window:

- When the output root directory is specified, the export directory of the file will be combined of the root output directory and the path typed in the **Name** field.
- In QMF for Workstation when the output root directory is not specified, the export directory of the file will be a combination of the application working directory and the path typed in the **Name** field.
- In QMF for WebSphere when the output root directory is not specified, the export directory of the file will be a combination of the temporal directory and the file name, the typed path is ignored.

6. Specify a file name for the file you want to create in the **Name** field. You can also specify the file extension in this field. If you do not specify it, the file type is selected from the list of available ones:

- **application/octet-stream** - *.bin
- **image/bmp** - *.bmp
- **image/png** - *.png
- **text/calendar** - *.ics
- **text/plain** - *.txt

Note: Specify a file extension if you want to create a file different from the listed ones.

7. In the **File Content** area, enter the content of the file you want to create in the **Content** field. For example, you want to add the text from **TextBox1**, enter the value: = `TextBox1.Text`.

8. Select the type from the **Content type** list or enter the any necessary value in the list. The available types for the content selection are:

- **application/octet-stream** specifies a binary file that is associated with an application that opens this file.
- **image/bmp** creates an image in *.bmp format.
- **image/png** creates an image in *.png format.
- **text/calendar** creates a document in iCalendar data format for representing and exchanging, calendaring and scheduling calendar information.

- **text/plain** creates an ordinary sequential file readable as textual material without much processing.
9. If you did not previously specify the directory parameter, select either **Global parameters** or **Local parameters** from the **Result parameter** drop-down list and click the **Add New Parameter button**.
The **Insert Parameter** window opens.
 10. Specify a necessary parameter and click **Finish**.
The **Insert Parameter** window closes and control returns to the **Add New Action** window.
 11. Select the parameter that you want to use from the **Result parameter** drop-down lists and click **OK**.
The Add New Action dialog closes.
 12. From the **[objectname] Behavior** window, click **Apply** to apply all actions that have been added for the event.
 13. Click **OK** to close the **[objectname] Behavior** window.

Adding a Refresh component event action

Use the **Refresh Component** event action to refresh a component in a visual dashboard by invalidating a query cache. This event action reruns the query and the results are displayed in the refreshed component.

About this task

The **Refresh Component** event action is appropriate when querying related components only.

To add a **Refresh component** event action to an object:

Procedure

1. Select the object in the editor window and double-click the event that will trigger the jump from the **Events** view.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** window opens.
3. Select **Refresh Component** in the actions tree.
Note: To quickly find the required event action, type the name of the action in the filter text field.
4. Click **Next**.
A window requesting the parameters for the **Refresh Component** event action opens.
5. Specify one or more target components to refresh and click **Finish**.
The **Add New Action** window closes.
6. Click **OK** to close the **[objectname] Behavior** window.

Results

You have added a Refresh Component event action.

The target component will be refreshed when the event action that you have specified occurs.

Adding an Invalidate query cache event action

Use the **Invalidate Query Cache** event action to invalidate any query cache, if the caching of result set data is enabled.

About this task

Using this event action is appropriate in a scenario where you have changed the target query table, thereby invalidating any previous results, and these invalid results are still displaying in the related components. By implementing the **Invalidate Query Cache** event action, you redraw results in all the related components. This query is rerun and the valid results are displayed in the component.

To add an **Invalidate query cache** event action to an object:

Procedure

1. Select the object in the editor window and double-click the event that will trigger the jump from the **Events** view.

The **[objectname] Behavior** window opens.

2. Click the **Add New Action** icon.

The **Add New Action** dialog opens.

3. Select **Invalidate Query Cache** in the actions tree.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. Click **Next**.

A window requesting the parameters for the **Invalidate Query Cache** event action opens.

5. Specify one or more target components for which the query cache will be invalidated and click **Finish**.

The **Add New Action** window closes.

6. Click **OK** to close the **[objectname] Behavior** window.

Results

You have added an Invalidate Query Cache event action.

When the defined event action occurs, query cache in the target component is invalidated and the valid results are displayed in the component the next time it is refreshed.

Adding a Launch LOB event action

Use the **Launch LOB** event action to open large object (LOB) data from an object of a visual dashboard.

About this task

LOB data is *binary* data of any type (for example, PDF files and the content of image files such as .gif or .jpeg files) that can be stored in a database table.

When run, the **Launch LOB** event action reads LOB values stored in a database table and performs an action associated with the value. The developer must specify the LOB *data type* in order for the application to launch the appropriate program and display the data.

Note: The **Launch LOB** event action can be applied to only those objects that are contained in the data template.

To add a **Launch LOB** event action, perform the following steps:

Procedure

1. Select the object in the editor window and double-click the event that will trigger the jump from the **Events** view.

The **[objectname] Behavior** window opens.

2. Click the **Add New Action** icon.

The **Add New Action** window opens.

3. Select **Launch LOB** in the actions tree.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. Click **Next**. A window requesting the parameters for the **Launch LOB** event action opens.

- a) Select **Data** from the drop-down list or type the necessary expression.

Note: The result of the expression must be the name of the column containing LOB data.

Select Data from the drop-down list or type the necessary expression. The result of the expression must be the name of the column containing LOB data.

b) Specify the data type by typing the necessary expression.

5. Click **Finish**.

The **Add New Action** window closes.

6. From the **[objectname] Behavior** window, click **Apply** to apply all actions that you have added for the event.

7. Click **OK** to close the **[objectname] Behavior** window.

Results

You have added a Launch LOB event action.

When the event action defined occurs, query cache in the target component is invalidated and the valid results will display in the component the next time it is refreshed.

When the visual dashboard runs you can open the LOB data by applying the specified event to the object that handles this action. In QMF for Workstation, the data opens in the associated system application. In QMF for WebSphere, the data opens in the web browser.

Working with modal dialogs in visual dashboards

You can design templates for modal dialogs and add them to your visual dashboards by using the Show dialog and Close dialog event actions.

About this task

To design a modal dialog and add it to a visual dashboard you need to perform the following actions:

Procedure

1. Create a dialog template to specify visual parameters of the dialog. You can also select a template from the existing ones or edit it.
2. Create a scene, which you want to use for a dialog.
3. In the scene properties, specify the dialog parameters and a dialog template.
4. Add controls, local variables, and event handlers to the dialog scene.
5. Add the Show dialog action to the main scene of the visual dashboard and specify its parameters and behavior.
6. Add the Close dialog action, and specify its parameters.

What to do next

See the following topics for more details about adding modal dialogs to visual dashboards:

Creating dialog templates

You must create a dialog template or select an existing one for a modal dialog that you want to add to your visual dashboard.

About this task

To create a dialog template:

Procedure

1. In the **Project Explorer** view, navigate to the **Globals** node of the visual dashboard to that you want to add a modal dialog.
2. Expand the **Globals** node to navigate to **Dialog Templates**. This node contains all available templates for the visual dashboard. You can select a pre-set one or create your own one:

- a) Right-click the **Dialog Templates** node and select **Insert Dialog Template** to create a dialog template.
 - b) In the **Dialog Templates** node, right-click an existing dialog template and select **Edit Dialog Template** to modify an existing template.
3. In the **Name** field, enter the name that you want to use for this dialog template.
 4. Specify the available properties for the template:
 - Content paddings
 - Background
 - Border
 - Shadow
 - Scene Background
 - Title Background
 - Title Font
 - Title Frame
 - Nine Patch
 5. When you specified the properties, click **Finish** to close the window and save the dialog template.

What to do next

See the following topics for more details about specifying the properties of a dialog template:

Specifying background properties for dialog templates

You can specify background properties for the content of a dialog template to customize its view.

About this task

To specify the background of the dialog template:

Procedure

1. In the **New Dialog Template Wizard**, select **Background** from the **Properties** list.
2. From the **Pattern** list, select the type of the pattern that you want to apply to the background.
3. In the **Color** field, specify the color for the background. Use the ellipsis (...) button to select a color from the palette.
4. Move the **Transparency** slider to set the transparency for the background color.
5. In the **Contrast color** field, specify the contrast color for the content borders. Use the ellipsis (...) button to select a color from the palette.
6. Move the **Transparency** slider to set the transparency for the background contrast color.
7. If you selected the gradient pattern, in the **Gradient type** list, select that gradient that you want to apply to the content.
8. If you want to set an image as a background, in the **Image source** field, click the ellipsis (...) button and specify the image that you want to use as a background.
9. If you use an image as a background, you can specify its location in the **Image anchor** list and tiling in the **Image tiling** list.
10. Click **Finish** to save the changes for this dialog template and close the dialog template window.

Specifying border properties for dialog templates

You can specify border properties for a data template to customize its view.

About this task

To specify border properties:

Procedure

1. In the **New Dialog Template Wizard**, select **Border** from the **Properties** list.
2. In the **Color** field, specify the color for the window borders. Use the ellipsis (...) button to select a color from the palette.
3. Move the **Transparency** slider to set the transparency for the border color.
4. Move the **Width** slider or enter a particular value to specify the width of the border.

Tip: You can customize the units of measure for borders by selecting a unit from the list of available ones.

5. In the **Top Border Radius** area, you can specify the fillet for the top border of the window. Move the **Top** left slider to set the fillet that you want to apply.
6. If you want to specify the fillet for the top right border separately, select the **Set top right border radius** check box and move the **Top right** slider.
7. In the **Bottom Border Radius** area, you can specify the fillet for the bottom border of the window. Move the **Bottom left** slider to set the fillet that you want to apply.
8. If you want to specify the fillet for the lower right border separately, select the **Set bottom right border radius** check box and move the **Bottom right** slider.
9. Click **Finish** to save the changes for this dialog template and close the dialog template window.

Specifying content paddings for dialog templates

You can specify content paddings for a dialog template to customize the offset between the dialog content and the dialog window borders.

About this task

To specify content paddings:

Procedure

1. In the **New Dialog Template Wizard**, select **Content** from the **Properties** list.
2. In the **Horizontal Paddings** area, move the **Left** slider to specify the left and the right padding of the content.
3. If you want to specify the right padding separately from the left one, select the **Set right padding** check box.

Tip: You can customize the units of measure for paddings by selecting a unit from the list of available ones.

4. In the **Vertical Paddings** area, move the **Top** slider to specify the top padding of the content.
5. If you want to specify the bottom padding separately from the top one, select the **Set bottom padding** check box.
6. Click **Finish** to save the changes for this dialog template and close the dialog template window.

Specifying NinePatch properties for dialog templates

Use the NinePatch images that contain the service information for correct image scaling and content placing.

About this task

To specify the NinePatch properties:

Procedure

1. In the **New Dialog Template Wizard**, select **Nine Patch** from the **Properties** list.
2. Select the **Use NinePatch image** check box.
3. In the **Title source** field, specify a path to the image for the title area.

- a) Click the ellipsis (...) button to open the **Picture Source** dialog.
- b) In the **Picture Source** window, select one of the following options:
 - **Linked** to specify that the image is in a local or network file. Select **Linked**, and specify the path to the image file.
 - **Embedded** to specify that the image is embedded in the visual project. In order for this option to be available you must have images available to the visual project for selection. All images that are available are listed in the **Images** folder that is under the **Globals** node in **Project Explorer**. To specify that the image is embedded, select **Embedded**. Select the image from the list of available images.
4. In the **Content source** field, specify a path to the image for the content area.
 - a) Click the ellipsis (...) button to open the **Picture Source** dialog.
 - b) In the **Picture Source** window, select one of the following options:
 - **Linked** to specify that the image is in a local or network file. Select **Linked**, and specify the path to the image file.
 - **Embedded** to specify that the image is embedded in the visual project. In order for this option to be available you must have images available to the visual project for selection. All images that are available are listed in the **Images** folder that is under the **Globals** node in **Project Explorer**. To specify that the image is embedded, select **Embedded**. Select the image from the list of available images.
5. Click **Finish** to save the changes for this dialog template and close the dialog template window.

Specifying scene background properties for dialog templates

You can specify background properties for the scene that hosts the dialog.

About this task

To specify background properties for the dialog scene:

Procedure

1. In the **New Dialog Template Wizard**, select **Scene Background** from the **Properties** list.
2. Follow the steps 2-10 in the [Specifying background properties for dialog templates](#) topic.

Specifying shadow properties for dialog templates

You can use a customized shadow for a dialog template.

About this task

To specify shadow properties for a dialog template:

Procedure

1. In the **New Dialog Template Wizard**, select **Shadow** from the **Properties** list.
2. Select the **Add a shadow** check box to apply a shadow to a dialog template.
3. Move the **Offset** slider to specify the offset of the shadow.
4. Move the **Angle** slider to specify the angle of the shadow.
5. Move the **Radius** slider to specify the radius of the shadow.
6. In the **Color** field, specify the color of the window shadow. Use the ellipsis (...) button to select a color from the palette.
7. Move the **Transparency** slider to set the shadow transparency.
8. Click **Finish** to save the changes for this dialog template and close the dialog template window.

Specifying title background properties for dialog templates

You can specify background properties for the dialog template title.

About this task

To specify the title background properties for the dialog scene:

Procedure

1. In the **New Dialog Template Wizard**, select **Title Background** from the **Properties** list.
2. Follow the steps 2-10 in the [Specifying background properties for dialog templates](#) topic.

Specifying title font properties for dialog templates

You can customize fonts in the titles of dialog templates.

About this task

To specify the font properties for a title:

Procedure

1. In the **New Dialog Template Wizard**, select **Title Font** from the **Properties** list.
2. From the **Font** list, select a font type for the title.
3. Move the **Size** slider to set the font size.
4. In the **Color** field, specify the font color for the dialog title.
5. Move the **Transparency** slider to set the transparency for the title font.
6. Select or clear the **Bold**, **Italic**, or **Underline** check box depending on the font type that you want to use.
7. Click **Finish** to save the changes for this dialog template and close the dialog template window.

Specifying title frame properties for dialog templates

You can customize the title frame in dialog templates.

About this task

To specify the title frame properties for a title:

Procedure

1. In the **New Dialog Template Wizard**, select **Title Frame** from the **Properties** list.
2. In the **Horizontal Margins** area, specify move the **Left** slider to specify the left and the right margins of the title.
3. If you want to specify the right margin separately from the left one, select the **Set right margin** check box.
Tip: You can customize the units of measure for margins by selecting a unit from the list of available ones.
4. In the **Vertical Margins** area, move the **Top** slider to specify the top and bottom margin of the title.
5. If you want to specify the bottom margin separately from the top one, select the **Set bottom margin** check box.
6. From the **Horizontal Alignment** list, select the alignment for the title.
7. Move the **Horizontal shift** slider to specify the title frame shift.
8. Click **Finish** to save the changes for this dialog template and close the dialog template window.

Adding the Show dialog event action

Use the Show dialog event action to add a modal dialog window that is based on a previously created scene to your visual dashboard.

Before you begin

You must have a previously created scene, which you want to add to the event action, and a dialog template, which is used as a container for the scene.

About this task

To add and define the Show dialog event action, perform the following steps:

Procedure

1. In the editor window, select the object to which you want to add the action.
2. Double-click the event that you want to trigger the action from the **Events** view.
The [objectname] Behavior window opens.
3. Click the **Add** button.
4. Select **Show Dialog** in the actions tree and click **Next**.
Note: To quickly find the required event action, type the name of the action in the filter text field.
5. In the **Add New Action** window, specify the action parameters.
6. If you want to open a scene from another visual project, select the **Another visual project** check box and specify a visual project.
 - a) Select the name of the scene that you want to add to the dialog window from the **Dialog scene** list.
 - b) If you have local parameters in the scene that you want to add from the current visual dashboard, you can modify their initial values for the destination scene when event occurs in the **Set values for the existing local parameters** list.
7. If you want to use a scene from the current project, perform the following steps:
 - a) Make sure that the **Another Visual Project** check box is clear.
 - b) Select the name of the scene that you want to add to the dialog window from the **Dialog scene** list.
 - c) If you have local parameters in the scene that you want to add from the current visual dashboard, you can modify their initial values for the destination scene when event occurs in the **Set values for the existing local parameters** list.
8. In the **Title** field, specify a title that you want to add to the dialog.
Note: The **Title** field value equals the **Title** parameter value of the scene that you use in the dialog.
9. In the **Dialog Location** area, specify the location of the dialog window. The available options are:
 - **Center location** to set the center of the scene as the location of the dialog.
 - **Custom location** to specify the exact coordinates for the dialog window.
10. Click **Next**.
The next page for specifying Show dialog action parameters opens.
11. Select the **Use Close Dialog action result** check box to use the parameter that is got in the Close dialog action.
12. If you did not previously specify a parameter for storing the result value of the event action, select either **Global parameters** or **Local parameters** from the **Result parameter** list and click the **Add New Parameter** button. The **Insert Parameter** window opens. In this window, specify a parameter and click **Finish**. The **Insert Parameter** window closes and control returns to the **Add New Action** window.

Note: You can add the parameters of the following types: Boolean, Integer, Double, Text, and Literal. Use the added parameter to specify conditions for the event actions execution. For example, if you use a Boolean parameter in the **DialogClose** event you can create two different actions that are run depending on the True or False returned value.

13. Specify the actions that you want to run when the dialog window closes. For the list of available actions, see the [Setting up event action navigation](#) topic.
14. Click the **Add** button. The **Add New Action** window opens. Specify the parameters for the selected action, and add it.

Note: The local parameters that are specified for the dialog scene are available in the **Event Parameters** category of the **Expression Designer**. If you have two parameters with the same name but with different values in the main and dialog scenes, the value that is specified for the dialog scene has higher priority and is used as the dialog scene parameter.

15. In the **Actions** list, you can modify parameters for the added actions.
16. Select the parameter that you want to use from the **Result parameter** lists and click **OK**.
The **Add New Action** window closes.
17. Click **Finish**.
The Show dialog event action is added. The **Add New Action** window closes.
18. From the **[objectname] Behavior** window, click **Apply** to apply all actions that were added for the event. Click **OK** to close the **[objectname] Behavior** window.

Results

You created a modal dialog window and added the Show dialog event action to the object.

Adding a Close dialog event action

Use the Close dialog event action to close a dialog window that is created in the Show dialog event action.

Before you begin

You must have a previously created modal dialog window.

About this task

To add and define the Close dialog event action, perform the following steps:

Procedure

1. In the editor window, select the object to which you want to add the action.
2. Double-click the event that you want to trigger the action from the **Events** view.
The **[objectname] Behavior** window opens.
3. Click the **Add** button.
The **Add New Action window** opens.
4. Select **Close Dialog** in the actions tree and click **Next**.
Note: To quickly find the required event action, type the name of the action in the filter text field.
5. In the **Add New Action** window, specify the return action parameter in the **Return value** field. You can use the specified value in the event actions that you add in the [Show dialog](#) event action.
6. Click **Finish**.
The Close dialog event action is added. The **Add New Action** window closes.
7. From the **[objectname] Behavior** window, click **Apply** to apply all actions that were added for the event. Click **OK** to close the **[objectname] Behavior** window.

Results

You added the Close dialog event action to the object.

Adding an Open URL event action

Use the **Open URL** event action to allow users to open hyperlinks from visual dashboards.

About this task

For example, you can use label with incorporated **Open URL** event action, by clicking on which users can go to the company's website. You can also use calculated expressions to create dynamic links that are formed on the basis of user's actions and the context in which they are used.

To incorporate an **Open URL** event action in an object:

Procedure

1. From the **Visual Designer** perspective, select the object in the editor window and double-click the event that you want to trigger the **Open URL** action.
2. In the **[objectname] Behavior** window, click the **Add New Action**.
3. Select **Open URL** in the actions tree and click **Next**.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. In the **URL parameters** area, specify whether to open the link in a new window or in a new tab.

Note: The **Target** parameter is actual only for QMF for WebSphere.

5. In the **URL** field, specify the link or calculated expression. To open the **Expression Designer** window, double-click in the **URL** field.

Note: You can specify either an absolute or relative URL. The relative URL starts with a slash (/) and contains the part of a link that succeeds the base URL. In this case, the absolute URL will be formed by concatenating the base URL (`http://host:port/application_path`) and relative URL.

6. Click **Finish**.

The **Print URL** action is added to the list of actions and the **Add New Action** window closes.

7. In the **[objectname] Behavior** window, click **Apply** to apply all actions that have been added for the event.
8. Click **OK** to close the **[objectname] Behavior** window.

Adding a Save Cookie event action

Use the **Save Cookie** event action to store the required data values under specific keys in the application storage for the current user.

About this task

In this event action, a user is an operating system user who works with the application or a user who runs the application in a web browser. Each user has a particular storage to keep the created key values. So the key values created by different users do not overwrite each other, even if the keys have the same name.

To add a **Save Cookie** event action to an object, perform the following steps:

Procedure

1. From the **Events** view, select the object in the editor window and double-click the event that you want to trigger.

The **[objectname] Behavior** window opens.

2. Click the **Add New Action** icon.

The **Add New Action** window opens.

3. Select **Save Cookie** in the actions tree and click **Next** to open the **Save Cookie action parameters** window.

Note: To quickly find the required event action, type the name of the action in the filter text field.

4. Specify the name of the key in the **Key** field.

5. Specify the value of the key in the **Value** field.
6. Click **Finish** to close the **Save Cookie action parameters** window and return to the **[objectname] Behavior** window.
7. Click **Apply** to apply all actions that have been added for the event.
8. Click **OK** to close the **[objectname] Behavior** window.

Results

The value of the key is stored in the application storage for the current user and can be retrieved by the name of the key any time.

Adding a Load Cookie event action

Use the **Load Cookie** event action to retrieve data previously saved in the **Save Cookie** event action from the application storage for the current user.

About this task

To add a **Load Cookie** event action to an object, perform the following steps:

Procedure

1. From the **Events** view, select the object in the editor window and double-click the event that you want to trigger.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** window opens.
3. Select **Load Cookie** in the actions tree and click **Next** to open the **Load Cookie action parameters** window.
Note: To quickly find the required event action, type the name of the action in the filter text field.
4. Type the name of the key which value you want to retrieve.
5. In the **Result option area**, click **Add New Parameter** to create a parameter for saving the value of the key.
Note: You can save the value of the key to the existing parameter by selecting parameter type and parameter name from the **Result parameter** lists.
6. Click **Finish** to close the **Load Cookie action parameters** window and return to the **[objectname] Behavior** window.
7. Click **Apply** to apply all actions that have been added for the event.
8. Click **OK** to close the **[objectname] Behavior** window.

Results

The value of the key is loaded to the parameter by the name of the key from the application storage for the current user.

Adding a Refresh geolocation event action

Use the **Refresh geolocation** event action to update the value of the Geolocation global parameter with the current geolocation data.

About this task

Important: The action is not supported in QMF for Workstation.

To add a **Refresh geolocation** event action, perform the following steps:

Procedure

1. Select the object in the editor window, and in the **Events** view, double-click the event that you want to trigger the **Refresh geolocation** event action.
The **[objectname] Behavior** window opens.
2. Click the **Add New Action** icon.
The **Add New Action** window opens.
3. From the **Device** tree node, select **Refresh geolocation**.
4. Click **Finish**.
The **Add New Action** window closes.
5. Click **OK** to close the **[objectname] Behavior** window.

Results

You added a Refresh geolocation event action.

Working with visual dashboards in version control systems

You can store visual dashboards in version control systems as folders. It allows you to track changes that are made by different visual dashboard designers, merge the corrections, or revert to earlier versions of visual dashboards.

About this task

To work with a visual dashboard in a version control system:

Procedure

1. Save a visual dashboard as a folder.
2. Add the folder with the visual dashboard files to a version control system.
Note: The `.lock` file is generated automatically to prevent one and the same visual dashboard from concurrent editing. Set the file to ignored when you work in a version control system.
3. After you edit the local version of the visual dashboard, save it and run synchronization to check and commit the updates.
4. After other users edit their local versions of the visual dashboard, update your local visual dashboard folder to check their corrections and, if necessary, save them to your local copy of the visual dashboard.
Note: Dashboards that are saved as folders are continuously edited in the course of their development and cannot be used for linking. If you need a link to a visual dashboard, for example when you add an **Embedded Scene** object, **Show Dialog** event action, or **Jump to New Location** event action, use only visual dashboards that are saved to the repository.

Saving visual dashboards as folders

You can save a visual dashboard as a folder with files that contain information about the content of a visual dashboard.

About this task

To save a visual dashboard as a folder:

Procedure

1. Create a visual dashboard or open the existing visual dashboard in the editor or in **Runtime** mode.
2. From the main menu, select **File > Save At > File** to open the **Save at File** dialog.
3. In the dialog, click **Visual Dashboard as Folder**.

4. In the **Path** field, specify the path to a destination directory to which you want to save a folder with the visual dashboard. Click **Browse** to navigate to the destination directory.

Note: Ensure that the folder to which you want to save the visual dashboard is empty.

5. Click **OK** to close the **Save at File** dialog and create the folder with visual dashboard files in the destination directory that you specified.

Displaying data on a map according to address information retrieved from a query

You can display data from a query result set on a map in your visual dashboards.

About this task

You use calculated columns and custom functions for calculated columns to convert addresses to coordinates in order to use them in your visual dashboards.

You must have created the *functions.js* file in the QMF for Workstation and QMF for WebSphere user home directory.

To display data on a map:

Procedure

1. Add the following function to the *functions.js* file:

```
/**
 * Returns address' geo coordinates
 * @param address address string
 * @category geo
 * @return lan/lng
 */
function getLocation(address)
{
    var request = new RSBIHttpRequest();
    request.open('GET', 'http://maps.googleapis.com/maps/api/geocode/json?address='
        + address + '&sensor=false', false);
    var response = '';
    request.onreadystatechange = function()
    {
        if(request.readyState === 4)
        {
            response = request.responseText;
        }
    }
    request.send(null);
    var x = JSON.parse(response);
    return x.results[0].geometry.location;
}
```

2. Create a visual query based on the required data table.
3. Add the *Coords* calculated column to the visual query with the following expression:

```
var loc = getLocation(@[address_column]);
('(' + loc.lat + ', ' + loc.lng + ')'
```

Save the visual query to a repository.

4. Create a procedure with the following text (replace the values in square brackets with the valid for your case):

```
RUN QUERY "[Path to the visual query]"
SAVE DATA AS [Table name] (ACTION=REPLACE, CONFIRM=NO
```

Save the procedure to the repository

5. Create new scheduled task that will run the saved procedure. Data will be updated on the defined scheduled task.

6. Create a visual dashboard displaying the data from the table on a map.

Protecting visual project data

You can compile read-only versions of visual reports and dashboards so that other developers can use them, but they cannot access and modify the underlying data objects.

About this task

To protect visual report and dashboard data:

Procedure

1. Open the previously saved visual report or dashboard which you want to compile in the design mode.
2. In the **Project Explorer** view right-click the project and select **Compile and Save As**.
3. From the **Save [Object]** wizard, select whether to save the visual project to a file or repository and click **Next**.
4. Provide the necessary information in the **Save to File** or **Save to Repository** wizard and click **Finish**.

Results

You have compiled a read-only version of your visual project. The data associated with the visual report or dashboard is protected and cannot be modified by other users.

Data source connections and visual projects

Visual projects (visual reports and dashboards) are intended to be shared across the enterprise. In order to facilitate sharing and the distribution of these visual projects, QMF for Workstation ties the query to a connection information alias (rather than associating the object to a specific data source).

You must set up a connection information alias for each data source that will provide data for these objects.

Specifying data source connections for visual projects

You set up a connection information alias for each data source that will provide data for the visual project (visual report or dashboard).

About this task

When you add the queries that will be used in a visual project, you must specify what connection information alias will be associated with each query.

To change the data source that will be used for a query, you edit the query information and select a connection information alias entry that points to the data source that you want to use. This feature allows users to easily share visual reports and use their own data sources. In addition, it facilitates moving between test and production data sources.

To specify a connection information alias for a data source, perform the following steps:

Procedure

1. Open the **Insert Connection** window.
You can open the **Insert Connection** window in one of the following ways:
 - From the **Project Explorer** view, right-click the **Connections** node for the specific visual project. Select **Insert Connection** from the pop-up menu.
 - From the **Project Explorer** view, double-click the **Connections** node. The **Insert Connection** window opens.

- Click a query from the **Workspaces** or **Repository Explorer** view. With the mouse button pressed, drag the query to the **Queries** node in the **Project Explorer** view. A new data source connection is automatically added to the **Connections** folder of the **Project Explorer** view.
2. From the list of available data sources, select the data source that will be associated with this connection information alias.
 3. Specify a unique name for this connection information alias in the **Connection name** field.
 4. Click **Finish**. The **Project Explorer** view lists the new connection information alias under the **Connections** node in the tree.

Note: You can also set up a connection information alias for a data source by dragging the data source from the **Workspaces** view to the **Connections** folder for the visual project in the **Project Explorer** view. The connection information alias is added with the name ConnectionN. You can edit the name of the connection information alias by right-clicking the entry and selecting **Rename**.

Results

You have set up the data source connection information for the visual data object.

Queries and visual projects

Visual projects (visual reports and dashboards) can include one or more queries to run to obtain the result data and display it in the project.

All queries that you use in the visual project must be defined in the **Queries** folder for that visual project.

You can add queries at any time while designing your project. For each query that you include in the visual project, you must specify what data source will be used. You do that by associating the query with a connection information alias.

Specifying queries for visual projects

Visual projects (visual reports or dashboards) can include one or more queries to run to obtain the result data and display it in the visual project.

About this task

You can create a new query that will be included in the visual project or you can use existing queries.

To specify a query for a visual project, perform the following steps:

Procedure

1. From the **Project Explorer** view, right-click the **Queries** node and select **Insert Query** from the menu.
2. Specify a unique name for the query in the **Query name** field.
3. Identify the query and specify where it resides in one of the following ways:
 - a) If you are creating a new visual query, select **Create a new visual query**.
 - b) If you are creating a new analytical query, click **Create a new analytical query**.
 - c) If you are creating a new forecast, click **Create a new forecast**.
 - d) If the query exists, select **Attach to an existing query**. The **Select query** list appears. Specify the name of the query and where it resides in the **Select Query** field. You can click **Browse** to search for the query.
 - e) If you would like to insert a static query, select **From file**.
 - Select **Insert Data** to copy the query directly into the **Queries** folder. This option provides greater portability, because the query will stay with the visual project if it is moved from one system to another.

- Select **Use link to file** to create a link to the query's system directory location. This option limits portability, because if the visual project is moved from one system to another, the query will no longer be available.
- Specify the query to be added in the **Select data file** list. You can click **Browse** to search for the query.

Note: You can open DBF, IXF, XML, and TAB files and use them as queries. You can edit the imported result set by filtering, adding calculated columns, applying grouping and aggregation. When you save the visual project, these files can be converted to dynamarts and saved with the visual project. All modifications will be preserved.

4. If you insert , you must select a data source connection that you want to use for the query from the **Connection** list. If the **Connection** list is empty, add a data source connection to a visual project.
5. Click **Add Connection** to open the **Insert Connection** window.
6. From the data sources tree, select the data source that you want to add for the visual project and click **Finish**.
The data source connection that you selected is added to the visual project and displayed in the **Connection** list.
7. If you are inserting an existing forecast or analytical query into a visual project, you can edit connection settings for subordinate queries in the **Edit Connections** window.
8. If you are inserting a new forecast or analytical query you cannot set connections for subordinate queries because these queries are not specified yet.
9. Click **Finish**. The **Project Explorer** view lists the query under the **Queries** node in the tree. If the query requires a parameter, a **Parameters** folder is added. You will automatically save the query when you save the visual project.

Results

Note: You can also add a query by dragging the query from the **Workspaces** view to the **Queries** node of the visual project in the view. The selected query is added to the **Queries** folder. A new data source connection is automatically added to the **Connections** folder of the **Project Explorer** view.

Adding custom JavaScript functions to visual projects

You can use your own JavaScript functions in visual reports and dashboards with the help of JavaScript Modules.

About this task

You use the JavaScript Module editor with syntax highlighting to write and edit your own functions. To open the editor, select **File > New > Other**. In the **New** wizard, select **QMF Objects > JavaScript Module**.

After you have created a JavaScript Module, you can save it in a repository or file system, and then add it to a visual report or dashboard.

To add custom JavaScript functions to a visual project:

Procedure

1. Open the **Insert JavaScript Modules** window. In the **Project Explorer** view, right-click the **Globals** node, and then select **New > Insert JavaScript Module** option.
2. In the **Insert JavaScript Modules** window, decide whether you need to create a JavaScript Module or attach an existing one.
3. If you decide to create a JavaScript Module, specify the required name in the **JavaScript Module name** field, select **Create new**, and then click **Finish**. The empty JavaScript Module opens in the separate editor window. Write the required JavaScript functions and close the editor.

4. If you decide to attach an existing JavaScript Module, select **Attach to an existing**, and then specify the required module from a repository or file system. Click **Finish**.

Note: You can view comments added in the JSDoc format to the script. These comments are visible to a user in the **Expression Designer**.

5. You can view the attached JavaScript Modules in the Project Explorer view under the **Globals > JavaScript Modules** node.

Results

All the functions defined in the attached JavaScript Module can be found in the **Expression Designer** under the **User** node. You can use them as other expressions in your visual project.

Creating JavaScript Modules

You create JavaScript Modules to customize JavaScript functions that you want to use in your visual projects.

About this task

To create a JavaScript Module, do the following actions:

Procedure

1. From the main menu, select **File > New > Other** to open the **New** wizard.
2. In the **New** wizard, expand **QMF Objects** folder and select **JavaScript Module**.
3. Click **Next** to open the **New JavaScript Module** wizard.
4. Specify the name for the JavaScript Module in the **Name** field and click **Finish** to open the **JavaScript Module** editor.
5. Enter JavaScript functions in the editor.
6. Start each function with the keyword 'function'.
7. Add a comment before each function.

The application interprets this comment as the description of the function.

8. Optional: Define a category to which you add a new function.

Use the '@category' tag in the comments to add it. If there is no such category, it will be created.

Note: The following categories already exist in the application:

- Columns
- Aggregation
- Conversion
- Date and Time
- Information
- Logical
- Math and Trigonometric
- Misc
- Text
- Objects

9. To verify the correctness of the JavaScript functions that you entered in the editor, click the **Check**

JavaScript syntax and structure  toolbar button.

10. After you entered the required JavaScript functions, save the JavaScript Module to a repository or a file.

Working with the Visual Designer

In QMF for Workstation, the **Visual Designer** perspective presents an editor window, several views and the menus that you will use to create visual projects.

About this task

You will use the following when you create either type of visual project:

- **Project Explorer**

This view displays in a tree structure all the elements and objects that will be included in the visual project.

- **Editor**

The editor window displays a canvas on which create your visual project. From the editor window, you can switch between different modes that allow you to design and check the final view of your visual project.

- **Properties view**

This view displays all the properties of each element and object that is included in the visual project. When you select an element from the **Project Explorer**, its properties are displayed in the **Properties** view. When you insert or select an object, its properties are displayed. When you make changes to an element's or object's properties, you change the way the element or object appears or behaves. You can make changes to properties directly in the **Properties** view.

- **Events view**

This view displays all the events available for each element and object in the visual project.

When you select an element from the **Project Explorer**, the events available for this object are displayed in the **Events** view. By using the **Events** view you can assign different event actions to scene objects and thus provide the scene interactivity and navigation at run time.

- **Palette view**

This view displays all of the graphic objects that you can insert in a visual project. You select objects from the **Palette** view and insert them in the editor. For each object that you insert, an entry is made in the **Project Explorer** and the properties for the object are displayed in the **Properties** view.

Working with the Project Explorer

The **Project Explorer** view is used when you are working with visual projects. It lists in a tree structure each visual report or visual dashboard that you currently have open.

About this task

As you add elements to your visual projects, each element is added to the **Project Explorer** tree as a folder. When you insert an object into your visual project, the object is also added to the **Project Explorer** tree as a folder. Depending on your element or object, additional folders are added.

The **Project Explorer** works in conjunction with the **Visual Designer** editor and the **Properties** view. When you select a visual project element from the **Project Explorer** all of the design objects that have been defined for the visual project element are displayed in the editor window and the properties for that element or selected object are displayed in the **Properties** view.

The **Project Explorer** allows you to copy, cut, and paste objects within a visual project or from one visual project to another. If you want to copy or cut multiple objects, make sure that all of them are stored within the same **Project Explorer** tree node. Objects from different nodes must be treated separately.

Each type of visual project includes different structural elements. Each visual report always includes the following high level folders:

- Connections

- Globals
- Queries
- Fixed pages
- Main page

Each visual dashboard always includes the following high level folders:

- Connections
- Globals
- Queries
- Scenes

As you add design objects to your visual projects, additional folders representing those design elements are automatically added to the **Project Explorer** tree.

Exporting images from visual projects

You can export images that are stored in a visual project to a graphic file.

About this task

Visual projects might contain different images that help to demonstrate the required information clearly. You can find these images in the **Project Explorer** view by expanding the **Globals > Images** node. If you need, you can export the images to graphic files.

To export an image to a file:

Procedure

1. In the **Project Explorer** view, select the required visual project.
2. To view the list of images, expand the **Globals > Images** node.
3. Right-click the required image and select **Save at file** from the menu.
4. To export all images that are stored in the visual project, right-click the **Images** node and select **Save at File** from the menu. The images will be packed in a ZIP archive.

Related tasks

[“Inserting picture objects” on page 417](#)

You use the **Picture** object to insert an image into your visual project.

Viewing object references

You can view all references to an object used in the visual project in the **Show References** window.

Before you begin

You must have a visual project opened in the **Visual Designer** editor.

About this task

To view object references:

Procedure

1. In the Project Explorer view, right-click the required object and select **Show References** from the pop-up menu.
The number of objects that have references to the selected object is displayed in the **Show References** window. If the selected object is referenced by other objects, the **Advanced** button becomes available.

2. To view all references, click **Advanced**, and then navigate to the object references in the tree.
3. You can use the arrows to navigate to the previous or to the following object that references the selected one.
4. To expand or collapse the tree of objects, click **Expand** or **Collapse**.
5. To view the selected referenced object in the **Project Explorer** view, click **Show in Project Explorer**.

Working with the editor

The **Visual Designer** editor window displays a canvas on which you create your visual project.

About this task

From the editor window you can switch between the modes that allow you to design your visual project and check how the project is displayed for users in different rendering modes.

You can open multiple visual project elements at the same time in the **Visual Designer** editor window. Each opened element appears as a separate canvas with an identifying tab along the top of the editor window. When you click the canvas tab in the editor window, that canvas becomes active. In order to make changes to a specific visual project element, that element must be active in the editor window. You can also activate a canvas by selecting the element of the visual project in the **Project Explorer** view.

The tabs at the bottom of the window allow you to toggle between the following modes:

- **Canvas mode**

You use **Canvas** mode to add content to your visual project. When you add objects they are represented in the editor by placeholder objects.

- **Design mode**

You use **Design** mode to arrange and modify objects, which you added in **Canvas** mode, to adjust the content of your visual project to different screen resolutions.

- **Runtime mode**

You use **Runtime** mode to test the content in your visual dashboard. In **Runtime** mode, graphic objects are drawn as they will appear to your users, queries are executed, object events are generated, and associated actions are performed. In addition, navigation features are executed. The **Runtime** mode of the editor displays your visual dashboard as the user will see it.

- **Preview mode**

You use **Preview** mode to test the content in your visual report. In **Preview** mode, graphic objects are drawn as they will appear to your users, queries are executed, object events are generated, and associated actions are performed. The **Preview** mode of the editor will display your visual report as a user will see it.

You can control how your visual project will be rendered in the **Runtime** and **Preview** modes by using the rendering mode list in the workstation menu bar.

Depending on the type of the visual project the following rendering mode options can be available:

- **HTML** optimizes the content to be viewed in a web browser.
- **HTML5** optimizes the content for better performance in modern web browsers and on mobile devices.
- **PDF** optimizes the content to be printed.

Editor controls in the visual designer

In the **visual designer** the editor rulers, guides, and grids help you to position visual objects precisely.

Rulers

When visible, rulers appear along the top and left side of the available window. The ruler origin also determines the grid's point of origin.

To show or hide rulers, select **View > Rulers**.

Pixels are used as the default unit of measurement. You can change the default unit of measurement on the **Visual Designer** page of the **Preferences** window.

To change the unit of measurement:

1. Select **View > Preferences**
2. Select **Visual Designer** from the tree.
3. Select one of **Visual Designer Ruler Unit** to specify the default units for the ruler. Your options are pixels, inches, or centimeters.

Guides

Guides appear as nonprinting lines above the scene and the objects it hosts.

To create a vertical guide, click somewhere on the horizontal ruler.

To create a horizontal guide, click somewhere on the vertical ruler. A new guide is drawn over the whole scene at the chosen position. You can also move the guides you created to new positions using the drag-and-drop method.

To remove a guide, select the guide and with the left mouse button pressed, drag it outside the ruler.

Grid

Use the grid for laying out elements symmetrically.

The grid is also drawn as nonprinting. By default, the grid displays whenever you open the **Visual Designer**.

To show or hide the grid, select **View > Grid**.

The grid's point of origin is located at the coordinates $0, 0$, which normally is at the center of the scene.

Note: The changes applied to the ruler's unit of measurement do not have any affect on the grid. Grid units always have a fixed size.

Snap to Grid

Use the **Snap to Grid** option to unbind objects and control their moving regardless to the visibility of the grid.

To move the scene components regardless to the visibility of the grid, select **View > Snap to Grid**. The objects are unbound and can be moved on the scene regardless to the grid.

Working with the Properties view

Each element and object that is included in a visual project has unique properties that dictate how the element or object will appear in the finished visual project. Properties for each element or object in a visual project are displayed in the **Properties** view.

About this task

Properties vary depending on the visual project element or object. They can include such things as location, value, size, alignment, line style, transparency, and color. They can also include properties that define how an object's appearance changes depending on a user's action.

Specifying object properties

Properties and their corresponding values for each element or object in a visual project are displayed in the **Properties** view.

About this task

The properties presented in the **Properties** view vary depending on the element or object that is selected in the editor window or in the **Project Explorer** view. Properties are listed with their current or default values.

There are many ways for you to modify the property values. You can modify the property values using constants, mathematical expressions, function expressions, or property expressions. For many properties, you can select a value from the property's list box of accepted values. Property values that you specify must match the data type required for the property.

To specify or modify individual object properties from the **Properties** view:

Procedure

1. The **Properties** view opens automatically when you are in the **Visual Designer** perspective. If the **Properties** view is not open, you can open it by selecting **Window > Show View > Properties**. To open the **Visual Designer** perspective, select **Window > Open perspective > Visual Designer**.
2. Select from the **Project Explorer** view, or the editor window, the element or object whose properties you want to view or modify. Once selected, the properties for that element or object are displayed in the **Properties** view.
3. Click the property cell for the property that you want to modify and enter the value. Depending on the property, the values that you enter could be a constant, a global or local parameter, a column name that is associated with the results of a query, a selection from a list of available values or an expression. The value you enter *must match* the data type required for the property.
4. You can use the **Expression Designer** to help you specify property values. Click the **Edit with Expression Designer** toolbar button in the **Properties** view to open the Expression Designer.
5. You can use the **Variables** bar to help you specify property values that will use global or local parameters or column names from a set of query results.

Specifying properties for groups of objects

You can specify properties for groups of objects that you have selected in the editor window.

About this task

To specify property values for a group of objects:

Procedure

1. In the **Editor** window, click the visual project canvas that is active.
2. With the left mouse button pressed, move your mouse to enclose the objects whose properties you would like to modify.
3. The common properties of all the objects that you selected are listed in the **Properties** view. If the values of a displayed property are the same for all selected objects, it will be displayed in the **Value** column. If any of the selected objects has a different value, the field is blank.
4. Click the property field for the property that you want to modify and enter the required value. All objects selected will be modified.

Using the Variables bar to specify property values

You can use the **Variables** bar to specify a property value. The **Variables** bar is located in the menu bar.

About this task

From the **Variables** bar, select one of the following options to set the object's property value:

- a global or local parameter that has been defined for the visual project.
- a column name from a set of query results.

To specify a property value using the **Variables** bar:

Procedure

1. In the **Variables** bar, select one of the following items from the first drop-down list:
 - **Global parameters** populates the second drop-down list with all the global variables that have been defined. Select the global variable that you want to use.
 - **Local parameters** populates the second drop-down list with all the local variables that have been defined. Select the local variable that you want to use.
 - One of the queries that has been defined for your project. The names for each of the columns that are included in the query results are listed in the second drop-down list. Select the column name that you want to use.
2. Click the **Link Variable to Object** icon. A red dot appears.
3. Pass your mouse over the red dot, and drag the dotted line that appears from the red dot to a property cell that is listed in the **Properties** view.
4. Click the property field. The column or parameter name is inserted as the value for the property that you selected.

Using the Expression Designer to specify a property value

The **Expression Designer** can be used to specify any property value that is associated with a selected object.

About this task

The **Expression Designer** is an alternative way of specifying an object's properties. Rather than type an object's properties directly in the property cell of the **Properties** view, you use the fields and buttons in the **Expression Designer** to help you create and format the property value. There are advantages to using the **Expression Designer** to specify your object's properties. For each property that can be specified for an object, you are given templates that can help you format the value of that property. In addition, you have more space available to work and view your property value (which can grow quite large if expressions are used).

To specify a property value using the **Expression Designer**:

Procedure

1. Select a property from the **Property** view. Click the **Edit with Expression Designer** toolbar button in the **Property** view. The **Expression Designer** window opens.
2. In the left pane of the window, each element of your visual project is listed in the **Expression Designer** tree. Expand each element to see a list of properties for the element.
3. Each of the built-in functions that can be included in property value expressions are listed in the right pane of the window. Expand the function categories to see what functions are available. You can filter the list of functions that are displayed using the filter field. You can type one or more letters in the filter field, and all the functions that start with the entered letter (or letters) are listed. You can also type in a specific function name and it will be listed.

4. The property that is being edited is displayed above the property value field. You can use both properties and functions to build your property value in the property value field. Double-click a function to insert it into the property value field. Double-click a property to insert a reference to it in the property value field. In both cases, the function or property is appended to what is already in the property value field. To replace content, highlight what you want to replace in the property field and then double-click the property or function. To insert content, position the cursor in the property field and then double-click the property or function. Functions are placed in the property field in template form. Depending on your logic, you will replace the parameters with your own values.
5. The following buttons, located below the property value box, provide additional help when creating a property expression:
 - **Undo:** Select this button to undo the last edit applied to the expression that you are building in the property value field.
 - **Redo:** Select this button to redo the last edit applied to the expression that you are building in the property value field.
 - **Edit:** Use this button to build an expression that will define a new property value. The property for which you are building an expression is displayed above the property value field. To change the property under edit, select the new property from the **Expression Designer** tree in the left pane of the window, and click the **Edit** button. The new property that will be edited is displayed above the property value field. You can set or edit multiple property values using the **Edit** button without closing the **Expression Designer** window. Each property that you modify is marked in the **Expression Designer** tree with an asterisk (*).
 - **Open FormatNumber dialog:** Select this button to open a list of available templates that you can use to format a number.
 - **Simplify expression:** Click this button to parse the expression that you have built in the property value field. The expression will be reviewed and simplified where possible.
 - **Select color:** Select this button to open the **Color** window. From the **Color** window you will select a color and click **OK**. The hexadecimal color number will be substituted for the highlighted color parameter displayed in the property value field.
6. Click **Apply**. The property value is updated.
7. Click **OK**. The property value is updated and displayed in the **Properties** view. The **Expression Designer** window closes.

Writing property expressions

Property expressions are conditional or mathematical expressions that you can specify as the value for an object's property. These expressions are evaluated to obtain the property value at runtime when a dashboard or report is displayed.

Property expressions enable you to create dynamic values for your object's properties. Many business decisions depend on factors such as market conditions, sales, operational expenses, new locations, profit and loss margins, and customer satisfaction. These factors are subject to change.

Elements of a property expression

Property expressions are composed of one or more elements that will supply values at runtime.

The following elements make up property expressions:

- **Functions:** Examples of functions include mathematical or statistical operations such as `Min()` and `StdDev()`. You can use functions to calculate values, convert values, or to specify Boolean conditions.
- **Operators:** Examples of operators include `=`, `>`, and `<`. You can use operators to perform arithmetic, to compare values, or to concatenate values.
- **Column Names:** Examples of column names might be any name that has been assigned to a column of data returned by a query. You can use these column names in your expressions.
- **Formatting symbols:** Examples of formatting symbols are `$`, `#`, and `%`. You can use these symbols to format numeric and date/time values.

- *Constants*: Examples of constants are values for any valid data type, such as True and False for a Boolean data type. Use constants in functions, with operators, with column names, and by themselves to specify a property value of the appropriate data type.

Required data types in return property values

When you specify an expression for a property value, the expression must return a value of the data type required by the property.

Object properties and their supported data types are listed by object in the Object Properties Reference. To return the value that you want in the data type required, you can convert it to the appropriate type using one of the built-in conversion functions.

Syntax rules for writing property expressions

You must use the correct syntax when you write property expressions.

The following table describes some basic rules for typing property expressions:

| Task | Rule | Example |
|--|--|---|
| Display a calculated value, such as a parameter or column name | Use equals sign (=) for calculated expressions | =cust_num |
| Display a literal value | Type the literal exactly as you want it to appear in text | Sales Organization |
| Display a text string in an expression | If the string is part of a combined expression that also includes calculated values, columns or parameters, it must be surrounded by quotation marks (" ") | ="My name is: "+fname |
| Concatenate two or more text strings | Use the plus sign (+) to concatenate text strings | =Click here to go to"+Dept+" department |
| Add two or more numeric values | Use the plus sign (+) to add constants or parameters of numeric data | =Param1+5 |
| Insert a carriage return in an expression | Type newline. If the expression elements are the same data type, use a plus sign (+) before and after. Otherwise, include newline in the Concat() function | =city+newline+state |
| Display the value of another property | Use the equals sign (=) plus the object name and the property name with dot notation | =Label1.Text |

Calculated expressions versus literals

All property expressions start with an equals sign (=). This indicates that the property must be computed rather than interpreted as a fixed series of characters.

The following examples show two property expressions for the **ToolTipText** property of a **Button** object:

Example 1

This example defines a literal value: 2+3

The tooltip will literally display the characters: 2+3

Example 2

This example defines a calculated value: =2+3

The tooltip will display: 5

Examples of property expressions

Some property expression syntax rules are best described using examples.

Writing an if() statement

You can use an `if()` function to create a Boolean (true/false) property value. For example, if you want to display the object as blue if the condition specified is met and yellow if it is not met, you might write the following expression:

```
Color =if(S_terrain="water", Blue, Yellow)
```

In this example, the condition of the object is based on a parameter, `S_terrain`. The object might be a polygon in a map layout or a data symbol shown on a chart. If the value of the scene parameter is `water`, the color of the object is blue. Otherwise, the color is yellow.

You can specify different conditions for the `if()` function, such as a column value or a numerical threshold. For example, you can make your visual project be responsive to changes and show their significance in the appearance of the objects in the project. Profit or loss, for example, can be displayed visually using color: black for profit, red for loss. The following property expression can be used to distinguish profit and loss in the color property of the object:

```
Color =if (sales>=expenses, Black, Red)
```

Displaying column data in a text string

You can enhance the data points for a layout object by adding a **Label** object that reveals information about the record returned. For example, you might want each (data point) in a tree chart to show the name of a department in a store, preceded by the word `Department`. You could insert a **Label** object into the data template for the layout object and then specify a property expression which combines the column name and a literal string.

```
Value ="Department: "+newline+dept_desc
```

In this expression, `dept_desc` is a column returned by the query to your data template.

Working with the Events view

Events are actions that are built into visual dashboards by designers. These actions provide the dashboard with all the necessary navigation and interactive functionality. The **Events** view displays these events.

The **Events** view consists of two panes.

The *upper pane* displays the events tree and the filter text field that you can use to quickly find the required event. The events tree contains all events that are available for the object that you select in a visual project. Each object has a default event that is displayed in the first root node of the tree. All other events that you can assign to the selected object are grouped by categories and displayed under the appropriate nodes of the tree.

The *lower pane* of the **Events** view includes the list of event actions defined for the selected event and instruments for their editing.

To open the **Events** view, perform the following steps:

1. From the main menu, select **Window > Show View > Other** and expand **Visual Designer Category**.
2. Select **Events** from the list of options and click **OK**.

When assigning an event to an object, designers can specify additional actions to be triggered when the event occurs. See the [Setting up event action navigation](#) topic for information about how to assign events to objects and how to specify the actions that are triggered when the event occurs.

For information about how to assign events and event actions to visual dashboards, see the [Assigning events and setting up event actions](#) topic.

Working with the Palette view

The **Palette** view displays all of the graphic objects that are available for insertion into your visual project.

About this task

The objects are grouped into the following *palettes* (also referred to as *drawers*):

Frequently Used

The **Frequently Used** palette contains 10 mostly used objects for easier navigation among them. A user can add objects to this palette or remove them in the **Customize Palette** dialog.

Note: To open the **Customize Palette** dialog, right-click any object on the palette and select **Customize**.

Primitives

The **Primitives** palette contains basic graphical objects that you can use to display text, rectangles, polygons, pictures and alignment panels.

Controls

The **Controls** palette contains graphical objects that you can use to display standard edit controls, such as sliders, buttons, and combo and list boxes. In addition, this palette includes the **Slicer** object that provides an easy way to filter the multidimensional data that is displayed in a layout object and the **DateTime** object that provides a convenient way for users to enter date and time information. The **Controls** palette is only available when creating visual dashboards.

Data Symbols

The **Data Symbols** palette contains graphical objects that are designed for inclusion in a layout object to augment a layout object's ability to represent its query results data. Data symbols can also be used independently of layout objects, and as such they are similar to the objects that are contained in the **Primitives** palette.

Layouts

The **Layouts** palette contains graphical objects that you can use to display the data that has been obtained as a result of a query. Layouts include a wide assortment of charts, graphs, maps, tables and grids.

Containers

The **Containers** palette contains the embedded scene object that you can use to link two scenes together and the embedded content object that you can use to add external content, such as a text file or HTML page, to a visual dashboard.

Miscellaneous

The **Miscellaneous** palette contains the Timer object, which is used to automate the occurrence of an event based on time intervals.

Connectors

The **Connectors** palette contains graphical objects that can be used to display a visual link between data points in a layout object or between two graphical objects that have been inserted in a visual report or dashboard. Connectors include an assortment of straight and curved lines.

Custom

The **Custom** palette contains objects from any of the other palettes. You are responsible for placing the objects that are on the **Custom** palette. You can place objects you have set custom properties for, or objects that you use frequently on the **Custom** palette. You place these objects in the custom palette so that they are easily accessible.

The **Palette** view also contains the tools that you can use when working with visual projects. You can find the following tools at the top of the **Palette** view:

Select

The **Select Tool** is used to select an object.

Grab and Pan

The **Grab and Pan** tools are used to move a dashboard scene around within the **Screen** editor window.

These tools are available when creating visual dashboards only.

Connectivity

The **Connectivity** tool provides users with the capability to define dependencies between dashboard control objects (such as combo or list box objects) and data-driven layout objects (such as a grid or bar chart object). You can also use the **Connectivity** tool to define dependencies between non-query driven objects such as text box objects and label objects.

The **Connectivity** tool is available when creating visual dashboards only.

Inserting palette objects into a visual project

From the **Palette** view, you can insert any graphical object that is listed into your visual project.

About this task

All objects from the **Palette** view can be inserted in a visual dashboard. Not all objects available from the **Palette** view can be inserted in a visual report. The objects that cannot be inserted in a visual report have interactive properties, and it does not make sense to insert them in visual reports because visual reports are not interactive.

To insert a graphical object into a visual project perform the following procedure:

Procedure

1. Open the **Palette** view.

Generally, the **Palette** view opens within the editor window when the visual project is active.

If the **Palette** view is not opened, open it using the following method:

- a. Select **Window > Open perspective > Visual Designer**
- b. With the **Visual Designer** perspective open, select **Window > Show View > Other**
- c. From the **Show View** window, select **General > Palette** and click **OK**

The **Palette** view opens.

2. In the editor window, select the visual report page or visual dashboard scene to which you want to add the object. If the page or scene is not open in the editor, then double-click the page or scene in the **Project Explorer**.
3. There are two ways that you can insert an object into the editor window:
 - From the **Palette** view, click the object that you want to insert into your visual project, then click in the editor window. A representation of the object is drawn on the canvas for the currently active visual project element that is in the editor window. The object is positioned in a default location and drawn to a default size. The properties of the object are displayed in the **Properties** view. The name of the object is added to the **Project Explorer** tree.
 - From the **Palette** view, click the object in that you want to insert into your visual project, then, click in the editor window, and drag diagonally across a rectangular area to define the location and size of the object. A representation of the object is drawn on the canvas positioned in the location that you have drawn at the size you have drawn. The properties of the object are displayed in the **Properties** view. The name of the object is added to the **Project Explorer** tree.
4. After inserting your object, you can modify the object's properties using the **Properties** view.
5. To alter the presentation of the objects as they appear in the **Palette** view, right-click anywhere in the **Palette** view to open a pop-up menu. Depending on how you want to view the objects in the **Palette** view, select one of the commands that are offered in the pop-up menu.

Inserting objects into a visual project by using the layout wizard

The layout wizard allows you to insert objects into your visual project by dragging the source query from the Repository Explorer tree to the editor area of your visual project.

About this task

To create an object by using the layout wizard, perform the following steps:

Procedure

1. In the **Repository Explorer** view, select the source query for your object and drag it to the editor area of your visual project.

The **Layout Wizard** opens.

2. On the **Select query role** page of the wizard, select one of the options and click **Next**.

- If you want to use your query as a source of data for a new object, select **Data Source**.
- If the query has several display modes and you want to use one of them as an object in your visual project, select **Layout**.

Depending on the option that you choose, the **Select the layout category and style** page or the **Select a chart or create one** page opens.

3. Select the object that you want to insert into your visual project and click **Next**.

Depending on the object, the corresponding wizard opens.

4. Complete the steps of the specified object's wizard to create the object.

For more information about creating particular objects, see the *Working with the Palette view* section of User Help.

Results

You inserted an object into your visual project.

Modifying the Palette view display settings

You can vary the way objects are displayed in the **Palette** view.

About this task

To vary the way objects are displayed in the **Palette** view:

Procedure

1. Open the **Palette** view.

Generally, the **Palette** view opens within the editor window when the visual project is active.

If the **Palette** view is not opened, open it using the following method:

- a. Select **Window > Open perspective > Visual Designer**
- b. With the **Visual Designer** perspective open, select **Window > Show View > Other**
- c. From the **Show View** window, select **General > Palette** and click **OK**

The **Palette** view opens.

Note: If there is no active visual project, the palette will be unavailable. To make the Palette available, you must first select a visual project. For example, a Visual Report or Visual Dashboard.

2. Each group of objects is displayed in a separate drawer. Clicking on the drawer will alternately expand and collapse the drawer, displaying or hiding the object icons.
3. Right-click anywhere in the **Palette** view. From the pop-up menu select one of the following options:

- **Layout → Columns** to arrange the object icons and their names in two columns.
 - **Layout → List** to arrange the object icons and their names in a list.
 - **Layout → Icons Only** to display only the object icons.
 - **Layout → Details** to display the object icons, their names and a description.
4. From the pop-up menu, select **Settings** to open the **Palette Settings** window where you can set the font that is used to display the name and descriptions of the icons, specify layout options, and specify drawer options:
- To specify font options, click **Change**. A **Font** window opens where you can change the default font.
 - To specify layout options, select from the **Layout** radio group. Also, select the **Use large icons** check box to display larger icons.
 - To specify drawer options, Select from the **Drawer options** radio group. Your choices are:
 - Always close when opening another drawer
 - Close automatically when there is not enough room
 - Never close
- Note:** You can also control the opening and closing of drawers by clicking the pin icon that appears in the drawer of each palette type. Clicking the pin open icon (it becomes highlighted) will cause the current drawer to stay open when you open another drawer. Clicking a highlighted pin will turn the pin open icon off and cause each drawer to be closed when another drawer is opened.
5. From the pop-up menu, select **Customize** to open the **Customize Palette** dialog.
- In the **Customize Palette** window, you can change the name and description of palette objects, rearrange how objects are displayed in the palettes, delete objects from the **Custom** palette, and control the visibility of palettes on start up.

Changing an object's default property value

Each object that you add to a visual project is inserted and drawn using the default values that have been set for the object's properties. You can set new default properties for these objects so they are drawn and inserted as you most frequently use them.

About this task

To change an object's default property value:

Procedure

1. In the **Palette** view, locate the object that you want to change.
2. Click the object to place it in the available editor window.
3. In the **Properties** view, change properties as you want.
4. When you are finished changing properties, return to the editor window, right-click the object and select **Set as Default** from the pop-up menu. The object's default property values are changed. Each time you select this type of object from the **Palette** view, it will be inserted and drawn in the editor window according to the new default properties.
5. To restore an object's original default properties right-click the object and select **Restore Default Properties** from the pop-up menu. If the menu item is unavailable in the **Palette** view, the object's default properties have not been changed.

Using the Connectivity tool

The **Connectivity** tool enables users to easily define dependencies between dashboard control objects (such as **Combo Box** or **List Box** objects) and data-driven layout objects (such as a **Grid** or **Bar Chart** object).

About this task

You can also use the **Connectivity** tool to define dependencies between non-query driven objects such as **Text Box** objects and **Label** objects. The **Connectivity** tool is only available when creating visual dashboards.

To connect a source object to a target object using the **Connectivity** tool:

Procedure

1. Create the source object. Select the object that will be the source object from the **Palette** view. Insert the object in your dashboard scene. In most cases the object that you select is one that can pass meaningful information to the target object. Typical source objects are control objects such as the **List Box** or **Combo Box** objects.
2. Create the target object. Select the object that will be the target object from the **Palette** view. Insert the object in your dashboard scene. In most cases you will select a target object that can change in some way after receiving meaningful information from the source object. Typical target objects are layout objects. When you create a layout object, you will specify the query that will supply the data for the layout object. The query should be placed in the **Queries** folder for the dashboard.
3. Click the **Connectivity** tool in the **Palette** view. Click the source object. A red dot appears. Click the target object.
4. If the target object is not a layout object, the source and target objects will be connected using a property value. The property value that is used is preset and based on the type of objects that are being connected. The **Connectivity** tool processing is complete. Review the properties for the target object as displayed in the **Properties** view. You will see the reference to the source object in the selected property cell. You can make manual changes to modify the property used to connect the two objects.
5. If the target object is a layout object, the source and target objects are connected using a parameter. You set up the parameter information in one of the following ways:
 - If the query for the target layout object does not include any parameters, then the **Select the column to be connected to the source object** page of the **Connection Options** wizard opens. Select the column from the **Columns** list that will be used to filter the query result data displayed in the target object based on the information in the parameter passed from your source object. Click **Finish**. You will be notified that a parameter has been added to the query for the layout object. Click **OK**. The two objects are connected. The **Connectivity** tool processing is complete.
 - If the query for the layout object already includes parameters, the **Select a connection type** page of the wizard opens. Click **Set an existing query parameter** to use one of the existing query parameters to pass connecting information from the source object to the target object. Click **Next**. The **Select a query parameter** page of the wizard opens. Select the parameter from the list of available parameters that have been defined for the query. Click **Finish**. The **Connectivity** tool processing is complete.
 - Click **Filter by an available table column** to select an available table column and create a new parameter. The **Select the column to be connected to the source object** page of the **Connection Options** wizard opens. Select the column from the **Columns** list that will be used to filter the query result data displayed in the target object based on the information in the parameter passed from your source object. Click **Finish**. You will be notified that a parameter has been added to the query for the layout object. Click **OK**. The two objects are connected. The **Connectivity** tool processing is complete.

Using the timer object

You use the **Timer** object to dictate or automate the occurrence of an event action based on time intervals.

About this task

You can find the **Timer** object in the **Miscellaneous** palette.

To insert a **Timer** object:

Procedure

1. In the editor window, select the visual dashboard scene to which you want to add the **Timer** object. If the scene is not open in the editor, then double-click the scene in the **Project Explorer** view.
2. From the **Palette** view, select the **Timer** object and click the editor window. The **Timer** object is drawn on the canvas in the editor window and it is added as a node in the **Project Explorer**.
3. From the **Properties** view, set the **EventInterval** property value. You will specify a time value in this property. The default is set to one (1) second.
4. You can create multiple **Timer** objects each having a different value for the **EventInterval** property. For example, one **Timer** object can have an **EventInterval** set to one second. Another **Timer** object can have an **EventInterval** set to five seconds.
5. In order for the **Timer** object to have any effect, you must set an event action to occur each time the **EventInterval** time elapses. For example, if you set the **EventInterval** property to one second, the event action that you assign to the specific **Timer** object will occur every one second.
6. To set an event action for the **Timer** object, select the **Timer** object from the **Project Explorer** view.
7. Double-click **Tick** from the **Events** view. The **[timerobjectname] Behavior window** opens.
8. Click the **Add New Action** icon. The **Add New Action** wizard opens. You can choose one of the following actions to occur on every tick of the **EventInterval**:
 - Jump to new location
 - Navigate to embedded scene
 - Note:** The **Navigate to embedded** scene action is only available to objects other than the **Embedded Scene** object if the **Embedded Scene** object is in the current scene.
 - Execute SQL statement
 - Note:** The **Execute SQL statement** action is only available to objects if one or more repository connections have been defined for the visual project in the **Connections** folder.
 - Execute procedure
 - Set values
 - Navigate back
 - Print scene
 - Execute shell command
 - Show message
 - Execute JavaScript
 - Export to Excel
 - Set focus
 - Reload scene
 - End session
 - Export to PDF
 - Send Email
 - Send Event to Embedded Scene

- Save Cookie
- Load Cookie

Although you can select any of the actions, the most common use of the **Timer** object is to control the behavior of another object. To control the behavior of another object at each tick of the **EventInterval**, select the **Set values** action. If you intend to control the behavior of another object, then you must have created the object before you start the process of assigning an action to the **Tick** event.

9. Click **Next**. The next window that opens depends on the action that you selected. If you selected the **Set values** action, the **Set values action parameters** window opens.
10. Double-click the **Parameter** cell. The **Expression Designer** opens.
11. The **Expression Designer** tree lists each object that is included in the selected visual dashboard. Expand the node for the object whose behavior will be changed at each tick interval. A list of the object's properties is displayed.
12. Select the property of the object that will change at each tick interval. For example, if the object is a **Label** object, select the **Text** property. Specify what the value of the text property will be at each tick of the interval using one or more of the available functions.
13. Click **OK**. At runtime, with a tick interval of one second, the content of the **Label** object will change to the new value that you specified every one second.

Adding primitive objects

Primitive objects are graphic objects such as text, lines, shapes, alignment panels, and pictures.

About this task

The following primitive objects are provided:

Label

Displays text, with or without a rectangular frame.

Line

Displays a straight line segment connecting two points.

Picture

Displays one of these image types:

- Graphics Interchange format (.gif)
- Joint Photographic Experts Group (.jpeg)
- Portable Network Graphics (.png)

Rectangle

Displays a rectangle.

Round Rectangle

Displays a rectangle with rounded corners. The radius of the curve at the corners is set in the Radius cell in the Properties view.

Ellipse

Displays a standard ellipse with foci aligned with the x- or y- axis.

Arrow

Displays a straight line segment with an arrowhead on one endpoint.

Double Arrow

Displays a straight line segment with an arrowhead on each endpoint.

Polyline

Displays a series of line segments connecting vertices. The vertices are specified using a PointSet data type.

Multiple Polyline

Displays one or more unique polyline objects that are connected or disjointed.

Polygon

Displays a series of line segments connecting vertices. The vertices are specified using a PointSet data type, where the final segment connects the first and last vertices.

Multiple Polygon

Displays a series of one or more disjointed polygon objects.

Alignment panel

Groups multiple objects relative to a single point so they all move in unison when based on a single location value.

Inserting text objects

You can use the **Label** object to insert text in your visual project.

About this task

To insert text into a visual project:

Procedure

1. In the editor window, select the visual report page or visual dashboard scene to which you want to add the object. If the page or scene is not open in the editor, then double-click the page or scene in the **Project Explorer**.
2. From the **Palette** view, select the **Label** object and insert it in the editor window.
3. From the **Properties** view, type the text that you want in the **Text** property cell. There are many additional properties that you can specify that will dictate the appearance of the text or specify how the **Label** object will behave on the visual report page or visual dashboard scene. There are several ways to specify property values for the **Label** object. You can select values that are listed in the **Properties** view, type values directly in **Properties** view, use the **Expression Designer**, or use the **Variables** toolbar.
4. For visual dashboards, you can define an event action for the **Label** object. This means that when a user performs a certain action on the object such as a mouse click, an event such as a jump to a scene will occur.

Results

Note:

- To format line breaks in text, insert +`newLine`+ where you want the break to occur.
- To create a border around text, display properties for the text object in the **Properties** view and select a line style in the **LineStyle** property cell.
- You can copy and paste multiple lines of text in the **Properties** view. The line breaks will automatically be displayed.

Inserting line objects

You can use the line objects to draw lines, arrows, double arrows, polylines or multiple polylines in your visual project.

About this task

To insert a line, arrow, double arrow, polyline or multiple polyline object into your visual project:

Procedure

1. In the editor window, select the visual report page or visual dashboard scene to which you want to add the object. If the page or scene is not open in the editor, then double-click the page or scene in the **Project Explorer**.
2. From the **Palette** view, select one of the following line objects:

- **Line:** You use this object to draw a single line. Select the **Line** object from the **Palette** view and click in the editor window (cursor has changed to a plus sign to represent drawing mode), drag the cursor to the point where you want the line to end, and release the mouse button. The line is drawn.
 - **Arrow:** You use this object to draw a single line with an arrowhead on the starting point. The **Arrow** object is actually a **Line** object with different default property values. It is drawn the same way a **Line** object is drawn.
 - **Double Arrow:** You use this object to draw a single line with an arrowhead on the starting point and ending point. The **Double Arrow** object is actually a **Line** object with different default property values. It is drawn the same way a **Line** object is drawn.
 - **Polyline:** You use this object to draw multiple lines that are connected at the end of each line. Select the **Polyline** object from the **Palette** view and click in the editor window (cursor has changed to a plus sign to represent drawing mode), move the cursor to the point where you want the first line to end, click the mouse button, move the cursor to the point where you want the second line to end, click the mouse button, continue until all lines are drawn. When you have finished drawing all the lines, hit the Esc (Escape) key. The cursor returns to an arrow (selection mode). All of the lines in the **Polyline** object are drawn.
 - **Multiple Polyline:** You use this object to draw multiple polylines. Select the **Multiple Polyline** object from the **Palette** view and click in the editor window (cursor has changed to a plus sign to represent drawing mode). Draw your first polyline and hit the Esc key. The cursor remains in the drawing mode. Draw your second polyline. Hit the Esc key. The cursor returns to an arrow (selection mode). Both polylines are drawn and are considered a single **Multiple Polyline** object.
3. From the **Properties** view, modify the properties, such as color, width, and style to draw your line as you want. There are many properties that you can specify that will dictate the appearance of the line object or specify how the line object will behave on the visual report page or visual dashboard scene. There are several ways to specify property values for the line objects. You can select values that are listed in the **Properties** view, type values directly in **Properties** view, use the **Expression Designer**, or use the **Variables** toolbar.
 4. For visual dashboards, you can define an event action for the line object. This means that when a user performs a certain action on the object such as a mouse click, an event such as a jump to a scene will occur.

Inserting shape objects

Primitive shape objects include polygons, rectangles, rounded rectangles, and ellipses.

About this task

To insert a shape object into your visual project:

Procedure

1. In the editor window, select the visual report page or visual dashboard scene to which you want to add the object. If the page or scene is not open in the editor, then double-click the page or scene in the **Project Explorer**.
2. From the **Palette** view, select one of the following shape objects:
 - **Rectangle:** You use this object to draw a rectangle. Select the **Rectangle** object from the **Palette** view and click in the editor window (cursor has changed to a plus sign to represent drawing mode), drag the cursor to draw the rectangle, and release the mouse button. The rectangle is drawn.
 - **Rounded Rectangle:** You use this object to draw a rectangle that has rounded corners. Select the **Rounded Rectangle** object from the **Palette** view and click in the editor window (cursor has changed to a plus sign to represent drawing mode), drag the cursor to draw the rectangle, and release the mouse button. A rectangle with rounded corners is drawn.
 - **Ellipse:** You use this object to draw an ellipse. Select the **Ellipse** object from the **Palette** view and click in the editor window (cursor has changed to a plus sign to represent drawing mode), drag the cursor to draw the ellipse, and release the mouse button. The ellipse is drawn.

- **Polygon:** You use this object to draw a closed, multi-sided figure. Select the **Polygon** object from the **Palette** view and click in the editor window (cursor has changed to a plus sign to represent drawing mode), move the cursor to the point where you want the first line of the polygon to end, click the mouse button, move the cursor to the point where you want the second line of the polygon to end, click the mouse button, continue until you have drawn the polygon shape. When you have finished drawing, select the Esc (Escape) key to terminate the drawing process. The cursor returns to an arrow (selection mode). The polygon figure is drawn.
 - **Multiple Polygon:** You use this object to draw multiple polygons. Select the **Multiple Polygon** object from the **Palette** view and click in the editor window (cursor has changed to a plus sign to represent drawing mode). Draw your first polygon and select the Esc key. The cursor remains in the drawing mode. Draw your second polygon. Select the Esc key. The cursor returns to an arrow (selection mode). Both polygons are drawn and are considered a single **Multiple Polygon** object.
3. From the **Properties** view, modify the properties, such as color, width, and style to draw your shape object as you want. There are many properties that you can specify that will dictate the appearance of the shape or specify how the shape object will behave on the visual report page or visual dashboard scene. There are several ways to specify property values for the shape object. You can select values that are listed in the **Properties** view, type values directly in **Properties** view, use the **Expression Designer**, or use the **Variables** toolbar.
 4. For visual dashboards, you can define an event action for the shape object. This means that when a user performs a certain action on the object (such as a mouse click) an event (such as a jump to a scene) will occur.

Inserting picture objects

You use the **Picture** object to insert an image into your visual project.

About this task

The image that you insert into a Picture object can have one of the following formats:

- Graphics Interchange format (.gif)
- Joint Photographic Experts Group (.jpeg)
- Portable Network Graphics (.png)

To insert a picture object into your visual project:

Procedure

1. In the editor window, select the visual report page or visual dashboard scene to which you want to add the object. If the page or scene is not open in the editor, then double-click the page or scene in the **Project Explorer**.
2. From the **Palette** view, select the **Picture** object and click in the editor window. The picture object is drawn based on default properties. You can also select the **Picture** object from the **Palette** view, click in the editor window, drag the cursor to draw the size that you want, and release the mouse button. The picture object is drawn.
3. You must specify where the image that will be displayed in the picture object is located. You can specify that the image will be found in a query result column, in a local or network file, or embedded in the visual project. From the **Properties** view, click the **ImageSource** property cell. Click the ellipsis (...) that appears in the cell. The **Picture Source** window opens. Select one of the following options
 - **Column** to specify that the image will reside in a query result column.

This option is available when a query is available to the visual project for selection. From the drop-down list of available queries, select the one that will return the result data. From the second drop-down list, select the column that will contain the image.

The Queries folder in the Project Explorer view lists all the queries that are available to the visual project

- **Linked** to specify that the image will reside in a local or network file. Select **Linked** and specify the path to the image file.
 - **Embedded** to specify that the image is embedded in the visual project. In order for this option to be available you must have images available to the visual project for selection. All images that are available are listed in the **Images** folder that is under the **Globals** folder in the **Project Explorer**. To specify that the image is embedded, select **Embedded**. Select the image from the drop-down list of available images.
4. From the **Properties** view, you can modify any of the additional properties to alter the appearance or behavior of the picture object on the visual report page or visual dashboard scene. There are several ways to specify property values. You can select values that are listed in the **Properties** view, type values directly in **Properties** view, use the **Expression Designer**, or use the **Variables** toolbar.
 5. For visual dashboards, you can define an event action for the picture object. This means that when a user performs a certain action on the object such as a mouse click, an event such as a jump to a scene will occur.

Results

Inserting an alignment panel object

The **Alignment Panel** object is an invisible reference object that you can use to group and position child objects relative to a single point.

About this task

When the single reference point of the alignment panel changes, the position of all the child objects assigned to the alignment panel changes as well. An alignment panel is most useful when you want a single data point in a layout object to be represented by several objects. For example, a scatter chart usually uses a single marker object for each data point that it displays. Using an alignment panel you could specify that for each data point in a scatter chart a marker, a text label, and two buttons will be displayed in the chart. Whenever your data point value changes and is displayed in a new location, all the objects in the alignment panel are displayed in the new location as well.

To create an alignment panel and use it in your visual project:

Procedure

1. In the editor window, select the visual report page or visual dashboard scene to which you want to add the object. If the page or scene is not open in the editor, then double-click the page or scene in the **Project Explorer**.
2. In the **Project Explorer**, select the element from the visual project to which you want to add an alignment panel.
3. From the **Palette** view, select the **Alignment Panel** object and insert it in the editor window. The **Alignment Panel** object is not drawn on the canvas in the editor window. It is added as a node in the **Project Explorer** view.
4. On the current canvas that is in the editor window, add the graphical objects that you want to be grouped together in the alignment panel. Each object is drawn on the canvas and inserted as a node in the **Project Explorer** view.
5. When you have finished adding each object, select the object name in the **Project Explorer**, and drag the object dropping it in the **Alignment Panel** node. Repeat this for each object that you want to be in the alignment panel. Upon completion, when you click any of the objects in the alignment panel, a box will surround the objects.
6. With the alignment panel selected in the **Project Explorer** view, select the **Location** cell in **Properties** view. Enter a value or column name. The center of the grouped objects will move to the coordinates that you specify.

Results

Note:

- In a layout object, the **Location** property of the **Alignment Panel** object takes different values, depending on the type of layout object that you are designing. For example, in a scatter chart, the data points are x- and y-coordinates based on columns from your query. To set the location of the alignment panel in a scatter chart, you specify the column names for the x- and y-coordinates. To determine what you should set the **Location** property for an alignment panel to, look at the **Location** property of the default data symbol for that layout object.
- If the alignment panel has a single value for its **Location** property, you can replace it with a single column from your query. If it has more than one value for its **Location** property, make sure that you replace each value with the appropriate column from your query, and that you retain the correct formatting for the **Location** property.
- For charts with axes that supply an alignment panel by default, the **Location** property for the alignment panel is set to the `Pt()` function which references one or more column names.

Adding control objects

Control objects are graphical objects that are used to display information or accept user input.

About this task

Each type of **Control** object has its own properties that make it suitable for a particular purpose. Using **Control** objects, you can request input from a user and based on the response to the control, you can trigger subsequent events. **Control** objects are only available for placement in visual dashboards.

The **Controls** palette offers the following control objects:

Text Box

An object that provides a box where users can enter and edit text.

Button

An object that when clicked by the user will trigger an action to be performed.

Check Box

An object that presents two choices to the user for selection. A check box control can be either selected or clear.

Radio Group

A composite object containing a set of radio buttons that present clickable choices to the user.

Horizontal Slider

A window that is used as a scrolling control for selecting an integer value within a range of values.

Vertical Slider

A window that is used as a scrolling control for selecting an integer value within a range of values.

Combo Box

An object that combines an editable text field with a list box. When displayed, the **Combo Box** object displays a box with the list of choices associated with the particular field. The user can enter additional choices using the editable text field of the **Combo Box** object. The **Combo Box** object can be used to filter the information that is displayed in the query-driven objects, such as **Grid** or **BarChart**.

List Box

An object that displays to the user a list of possible values that can be selected for a particular field. The **List Box** object can be used to filter the information that is displayed in the query-driven objects, such as **Grid** or **BarChart**.

Date Time

An object that makes it easy for users to enter date or time information which can then be used by other objects.

Tree

An object that displays data objects in a visual dashboard and allows to drag these objects to the **Canvas** object.

Prompt Hierarchy

An object that allows to display the appropriate part of query results by selecting the levels and elements of a prompt hierarchy that is associated with the query.

Inserting a control object

Control objects are graphical objects that are used to display information or accept user input.

About this task

Each type of **Control** object has its own properties that make it suitable for a particular purpose. Using control objects, you can request input from a user and based on the response to the control, you can trigger subsequent events. **Control** objects can only be inserted in visual dashboards.

To insert a **Control** object into a visual dashboard:

Procedure

1. In the editor window, select the visual dashboard scene to which you want to add the object. If the scene is not open in the editor, then double-click the scene in the **Project Explorer**. If you are inserting a control to a **Level** folder of a visual dashboard, then select that **Level**.
2. From the **Palette** view, select the control object from the **Controls** palette and insert it in the editor window.
3. From the **Properties** view, modify the object's properties to make the control object appear and behave as you want. There are several ways to specify property values for the control objects. You can select values that are listed in the **Properties** view, type values directly in **Properties** view, use the **Expression Designer**, or use the **Variables** toolbar.
4. You can connect the control object to a target object, in most cases that is a layout object. This allows you to change the behavior of a target layout object based on the user's input to the control object.

For example, you can connect a **List Box** control object with the department names of your company to a **Grid layout** object containing all the department information. Use the Connectivity tool to connect the two objects. Your user will select a department from the list box, and only the information for that department will be displayed in the grid.

5. You can define an event action for the control object. This means that when a user performs a certain action on the object such as a mouse click, an event such as a jump to a scene will occur.

Inserting Combo Box and List Box objects

Combo Box and **List Box** are the objects that you can use to control the data that is displayed in your visual dashboard, or to accept the user input. You can link both objects to query driven layout objects.

About this task

To add a **Combo Box** or a **List Box** object to your visual dashboard, complete the following steps:

Procedure

1. In the **Palette** view, expand **Controls**, click **Combo Box** or **List Box** object, and place it on your visual dashboard.
The **Combo Box Wizard** or the **List Box Wizard** opens.
2. In the wizard, specify the way you want to populate the OptionList and the ItemDataList of your **Combo Box** or **List Box** object with data.

Note: The OptionList items are displayed in the **Combo Box** or the **List Box** object. The ItemDataList items are not displayed in the object. Each item of the ItemDataList corresponds with an item of the OptionList and can be used for linking objects by hidden value. Both OptionList and ItemDataList can

be used to link the **Combo Box** or **List Box** object to query-driven layout objects (such as **Grid** or **BarChart**). For more information, see the [Linking Combo Box and List Box objects to query driven objects](#) topic.

You can supply the data for both lists from a query result set, a static list, or an expression.

- To populate the list with data from a query, perform the following steps:
 - a. On the first page of the wizard, select **Yes** and click **Next**.
 - b. Add the source query and click **Next**.
 - c. From the **Field containing the list of options to be displayed** list, select the query column the values of which you want to include in the OptionList.
 - d. From the **Field containing the item data (optional)** list, select the query column the values of which you want to include in the ItemDataList.
 - e. Click **Finish**. The wizard closes.
- To supply the OptionList items from a static list, perform the following steps:
 - a. On the first page of the wizard, select **No** and click **Next**.
 - b. Click **Add New Entry**.
 - c. Add items to the list.
 - d. Double-click one or several check boxes in the **Value** column to make the corresponding items selected by default. For the **Combo Box** object, you can select one item from the list. For the **List Box** object, you can select one or several items from the list. Click **Finish**. The wizard closes.
- To supply the ItemDataList items from a static list, perform the following steps:
 - a. On the first page of the wizard, select **No** and click **Next**.
 - b. Click **Finish**. The wizard closes.
 - c. On your visual dashboard, click the object that you just created to select it.
 - d. In the **Properties** view, click the **ItemDataList** property cell and click the ellipsis (...). The **List Editor** window appears.
 - e. In the window, add items to the list and click **OK**.
- To supply the list items from an expression, perform the following steps:
 - a. On the first page of the wizard, select **No** and click **Next**.
 - b. Click **Finish**. The wizard closes.
 - c. On your visual dashboard, click the object that you just created to select it.
 - d. In the **Properties** view, click the **OptionList** or **ItemDataList** property cell and type the expression or double-click the property cell to use the **Expression Designer**. Consider the following expression as an example: `=lineInsert(Parameter1, 1, TextBox1.Text)`. In this example, `Parameter1` is the global parameter of the “TextSet” type of your visual dashboard, `TextBox1.Text` is the string that you want to put on the list, and `1` is the position of that string in the list. For more information about using the **Expression Designer**, see the [Using the Expression Designer to specify property or field values](#) topic.

Results

You added the **Combo Box** or **List Box** object to your visual dashboard. Use the **OptionList** property, the **ItemDataList** property, or the column from the result set of the query that your **Combo Box** or **List Box** object is based on to link your **Combo Box** or **List Box** object to query-driven layout objects (such as **Grid** or **BarChart**).

Linking Combo Box and List Box objects to query driven objects

Use the **Connectivity** tool to link your **Combo Box** or **List Box** object to query driven layout objects, such as **Grid** or **BarChart**.

Before you begin

Before you can link objects, you must populate the **OptionList** and the **ItemDataList** of your **Combo Box** or **List Box** object with data.

About this task

To link your **Combo Box** or **List Box** object to a query driven layout object, perform the following steps:

Procedure

1. In the **Palette** view, click the **Connectivity** tool.
2. On the visual dashboard, click your **Combo Box** or **List Box** object.
3. Click your layout object. The **Connection Options** wizard opens.
4. In the wizard, specify the source query column that you want to use to drive your destination object.

To link the objects, you can use the column from the **OptionList** property, the column from the **ItemDataList** property, or any other column from the result set of the query, that your **Combo Box** or **List Box** object is based on. If you use the column from the **OptionList** property, you link the objects by values that are displayed in your **Combo Box** or **List Box** object. If you use the column from the **ItemDataList** property, you link the objects by hidden values which are different from the ones that are displayed in your **Combo Box** or **List Box** object. If you use the column from the result set of the query, that your **Combo Box** or **List Box** object is based on, you link the objects by the values that are stored in that column.

If you want to link your objects by values from the **OptionList** property or the **ItemDataList** property, make sure that you specify one and the same column name in the **Connection Options** wizard and in the **OptionList** or the **ItemDataList** property of your control object.

Click **Next**.

5. Select the target query column that you want to connect to your **Combo Box** or **List Box** object. Click **Finish**.
6. In the notification window, click **OK**.

Results

You linked the objects. The information displayed in your target layout object now depends on the item selected in your **Combo Box** or **List Box** object.

Inserting a Slicer object

A **Slicer** object is used to filter the multidimensional data that is displayed in a layout object.

About this task

To insert a **Slicer** object into a visual dashboard and set up a connection to another layout object:

Procedure

1. In the editor window, select the visual dashboard scene to which you want to add the object. If the scene is not open in the editor, then double-click the scene in the **Project Explorer** view.
2. From the **Palette** view, select the **Slicer** object from the **Controls** palette and insert it in the editor window. The **Slicer Wizard** opens.
3. On the first page of the wizard select the OLAP query of the layout object that you would like to filter.
4. Click **Next**.

The **Select dimension or hierarchy** page opens.

5. Select the dimension of the query that you would like to filter and click **Finish**.
The slicer object is inserted in the editor window.
6. Select the **Connectivity** tool from the **Palette** view and click the slicer object.
A red dot appears.
7. Click the layout object that you want the slicer object to filter.

Results

At runtime (when the dashboard is displayed to the user), the user can click the **Slicer** object, and the **[Dimension Name] Filter** window opens. From the **[Dimension Name] Filter** window, the user can check which hierarchies or hierarchy elements that will be displayed in the layout object and click **OK**. The layout object refreshes to show only the selected data. For example, if the slicer object is set to filter the product dimension of the layout object's OLAP query, the user can choose to only display totals for a particular product.

Inserting a Date Time control object

A **Date Time** control object is used to obtain date or time information from a user. The user specified date and time information can then be used to control the behavior of another object.

About this task

To insert a **Date Time** object into a visual dashboard and set up a connection to another object:

Procedure

1. In the editor window, select the visual dashboard scene to which you want to add the object. If the scene is not open in the editor, then double-click the scene in the **Project Explorer** view.
2. From the **Palette** view, select the **Date Time** object from the **Controls** palette and insert it in the editor window.
3. In the **Property** view, select the value for the **Mode** property of the **Date Time** object.
 - Select the **Date** mode to specify that the control object accepts the date value as input.
 - Select the **Time** mode to specify that the control object accepts the time value as input.
 - Select the **Date Time** mode to specify that the control object accepts the date and time values as input.
4. Insert the object whose behavior will be controlled by the value entered in the **Date Time** object. You can insert an object that will display the return value of the **Date Time** object or you can insert an object that will use the **Date Time** return value as a parameter.
5. To display the **Date Time** object return value:
 - a) Insert a **Label** object.
 - b) From the **Properties** view, select the **Text** property for **Label** object.
 - c) Open the **Expression Designer**.
 - d) In the left pane of the **Expression Designer**, expand the **Date Time** object whose return value you want to display. Double-click the **Value** property.
 - e) The `Label1Object.Text` property will equal the `DateTimeObject.Value`.
6. To use the **Date Time** object return value as a parameter:
 - a) Create a query object (Query1) that queries a table that contains a date column. The query should include a parameter. For example: `Select * from q.interview where INTDATE=&dateparameter`. When you save this query (or move the query) to the **Queries** folder for the dashboard, a parameter with the same name as the substitution variable is created. In this example, a parameter called `dateparameter` is created.
 - b) From the **Project Explorer**, expand the **Queries** folder, expand the Query1 folder, and right-click the parameter. Select **Edit Parameter** from the pop-up menu. When parameters are created based on

their existence in a query, they are created with a datatype of literal. Change the datatype of the parameter to **Date Time**. Set a default value for the parameter. Click **OK**.

- c) Set the query parameter value to equal the return value of the **Date Time** object. From the **Project Explorer**, select the **DateTime** object whose return value will be used to set the parameter value. From the **Properties** view select the **Value** property for the **Date Time** object. Open the **Expression Designer**. In the left pane of the **Expression Designer**, expand the **Queries** folder for the dashboard. Expand the Query1 query. Expand the Parameters folder. Double-click the dateparameter parameter. The DateTimeObject.Value property will be set to equal the dateparameter.
- d) Insert a **Grid** object whose associated query is Query1. Query1 uses the dateparameter to control what rows are included in the grid. In this example, the dateparameter value is supplied from the **Date Time** object **Value** property.
- e) You can also use the **Connectivity** tool to set up the connection between the **Date Time** object and the **Grid** object.

Inserting a Tree control object

A **Tree** object provides an additional mechanism that you can use to display objects in a visual dashboard and drag objects from a **Tree** to a **Canvas** object.

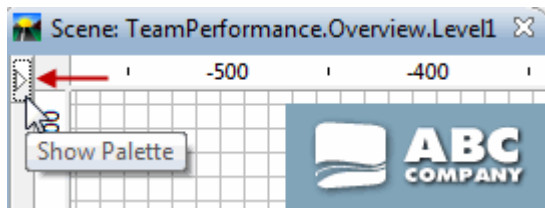
About this task

To insert a **Tree** object to a visual dashboard:

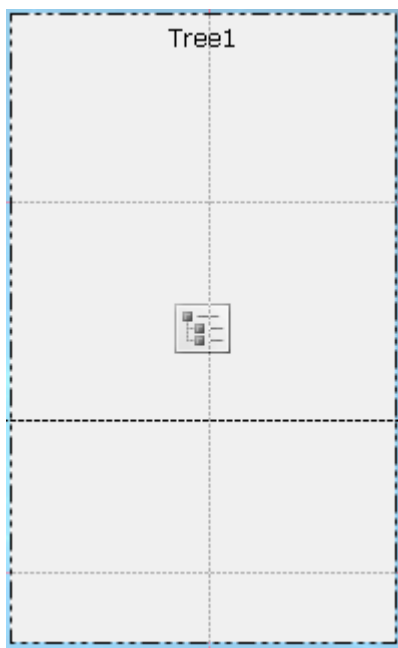
Procedure

1. Make sure the visual dashboard is opened on the **Canvas** tab of the workbench editor area.
2. Click the **Show Palette** icon to show the list of folders that hold the elements that you can add to your visual dashboard.

Note: The **Show Palette** icon is located in the upper-right corner of the editor area:



3. From the **Controls** palette of the **Palette** view, select the **Tree** object and insert it in the editor window. The **Tree** object is added to a visual dashboard. By default, the Tree object is named Tree1 and appears as *grey box with a dotted line perimeter* in the editor area:



Results

The **Tree** object provides a view into the hierarchical structure of data visualization objects and data transformation objects.

To view the hierarchical structure of the Tree, switch the editor to **Runtime** mode.

What to do next

In **Runtime** mode you can drag the visual dashboard objects from the **Tree** object to the **Canvas** object.

Inserting a Prompt Hierarchy control object

The **Prompt Hierarchy** object is a control that allows the user to select one or more values from a flat or hierarchical list of options to be displayed.

About this task

Prompt Hierarchy control objects allow a user to control what hierarchy levels and individual elements of a prompt hierarchy are expressed in query results displays. For example, you create a table that is populated by a query that has an associated prompt hierarchy. You then create a Prompt Hierarchy control and link it to the query's prompt hierarchy. Now, when you run the dashboard, you can open a window and select which parts of the query data that you want to display in the table.

To insert a Prompt Hierarchy control object:

Procedure

1. In the editor window, select the visual dashboard scene to which you want to add the object. If the scene is not opened in the editor, then double-click the page or scene in the **Project Explorer**.
2. From the **Palette** view, select the **Prompt Hierarchy** object and click in the editor window or double-click the **Prompt Hierarchy** object. The **Prompt Hierarchy** object is drawn based on default properties.

Note: You can also select the **Prompt Hierarchy** object from the **Palette** view, click in the editor window, drag the cursor to draw the size that you want, and release the mouse button. The **Prompt Hierarchy** object is drawn.

3. Select the query that you want to connect to the control object from the **Query** drop-down list.

Note: All of the queries that have been previously added to the **Queries** folder in the **Project Explorer** are available in the **Query** drop-down list.

4. Select the prompt hierarchy that you want the object to control from the **Prompt hierarchy** drop-down list.

Note: All of the prompt hierarchies that have been previously linked to the queries in your **Queries** folder in the **Project Explorer** are available in the **Prompt hierarchy** drop-down list. If no prompt hierarchies have been associated with your dashboard queries, this drop-down list is empty.

5. Click **Finish**.

The **Prompt Hierarchy Wizard** closes and the new Prompt Hierarchy control object is displayed on the canvas.

Adding data symbol objects

A **Data Symbol** object is an icon that is used to represent a data point in a layout object.

About this task

Each graphical layout object includes a default data symbol object which is used within the layout object to display the data. You can use a data symbol object to change a default data symbol in a layout object, to create a second data layer in a layout object, or independently, as a direct insert to a visual project (perhaps to create a legend for a layout).

The **Data Symbols** palette includes the following data symbol objects:

Google Map Marker

A data symbol object that displays a data point as a map marker on a Google Map object.

Google Map Marker is available in visual dashboards only.

Marker

A data symbol displayed as a configurable shape. The marker is typically used to represent a data point in an axis-based chart. The shape of the marker can be changed to a solid diamond, a cross, or any of several other shapes.

Vertical Value Bar

A data symbol displayed as a vertical rectangle in which the height is associated with a data value returned by a query. The vertical value bar is typically used to create bar values along the y-axis of an XY chart.

Vertical Label Bar

A data symbol displayed as a rectangle that is described by a label that places it along the x-axis and a numeric value that determines its height along the y-axis.

Horizontal Value Bar

A data symbol displayed as a horizontal rectangle in which the width is associated with a data value returned by a query. The horizontal value bar is typically used to create bar values along the x-axis of an XY chart.

Horizontal Label Bar

A data symbol, displayed as a rectangle that is described by a label that places it along the y-axis and a numeric value that determines its length along the x-axis.

Price Interval Bar

A data symbol displayed as a vertical line with horizontal ticks extending to the left and right side of the line. It is used to display four numeric values over a specific time interval. The price interval bar is typically used to display the opening, closing, high, and low prices of a stock or other index on a specific date. The shape of the price interval bar can be modified to turn open/close and high/low tick marks on or off.

Candlestick

A data symbol displayed as a vertical rectangle. Used to display four numeric values over a specific time interval. The candlestick is typically used to display the opening, closing, high, and low prices of

a stock on a specific date, and the color of the candlestick is designed to show gain or loss of stock value.

Event Band

A data symbol displayed as a rectangle of variable width extending from the bottom to the top of the chart. It is used to display ranges of x-values, such as a length of time an event lasted, along an axis. The event band is typically used with other data layers to depict events that correspond to other data.

Threshold Band

A data symbol displayed as a horizontal rectangle. The threshold band is similar to the event band data symbol. It is used to display ranges of y-values along an axis and is typically used with other data layers to depict value thresholds that correspond to other series of data.

Inserting a data symbol object

A data symbol is an icon that is used to represent a data point in a layout object.

About this task

Each graphical layout object includes a default data symbol object which is used within the layout to display the data. You can use a data symbol object to change a default data symbol in a layout object, to create a second data layer in a layout object, or independently, as a direct insert to a scene (perhaps to create a legend for a layout).

To add a data symbol to a visual project:

Procedure

1. In the editor window, select the element of your visual project to which you want to add the data symbol object. If the page, scene or object is not open in the editor, then double-click the page, scene, or object in the **Project Explorer**.
2. From the **Palette** view, select the data symbol object and insert it in the editor window.
3. From the **Properties** view, modify the object properties to make the data symbol appear the way you want it to.

Results

Note:

- If you want more than one object to represent each data point, first insert an alignment panel into the data template and then place the data symbol in the alignment panel.
- Adding an event band data symbol in a data layer adds depth and significance to a timeline chart. You might do this if you wanted to display particular event instances on a timeline chart that tracks a continuous trend in the market.

Adding connector objects

A connector is a graphic object, such as a straight or curved line, that provides a visual link between data points in a layout object or between graphic objects on a report page or dashboard scene.

About this task

If applicable, a layout object has a default connector which is used to provide a visual line between the data points in that layout. You can customize the line that connects all the data points in a layout by selecting a new connector from the **Connectors** palette. From the **Connectors** palette, you can also add a connection point when you want to force all data points or objects to terminate at one specified point.

The **Connectors** palette offers the following connectors:

• Straight Connector

A connector between two data points or two objects that is displayed as a straight line.

• Elbow Connector

A connector that creates a link between two data points or two objects by using both horizontal and vertical lines connected with right-angle joints.

- **Spline Connector**

A connector between two data points or two objects that is displayed as a smoothly curved line.

- **Connection Point**

An invisible point whose coordinates are used as a reference for a link extending from one data point to another data point.

Inserting a connector object

A connector is a graphic object, such as a straight or curved line, that provides a visual link between data points in a layout object or between graphic objects on a report page or dashboard scene.

About this task

To insert a connector into a visual project:

Procedure

1. In the editor window, select the visual report page or visual dashboard scene to which you want to add the object. If the page or scene is not open in the editor, then double-click the page or scene in the **Project Explorer**. If you are inserting a connector into a layout, select the **Layout** and the **Level** from the **Project Explorer**.
2. From the **Palette** view, select the connector object and insert it in the editor window.
3. From the **Properties** view, modify the properties to draw your line as you want.
4. Click **Runtime** or **Preview** to view the results.

Tip:

- To force all connection lines to connect at a single specified point, insert a **Connection Point** and position it as you want.
- Hierarchy layouts and chart layouts with line connectors contain implicit connection points. No **Connection Point** object appears in the **Project Explorer** when these Layouts are created. However, there is a Connection Point property defined for it.
- The organization chart object uses a connection point to override the point-to-point links by making all data points connect to a single parent data point.

Example

Suppose you want to create a decision tree using circles to represent decisions and multiple lines to link decisions to multiple choices. You might create an organization chart and then replace the boxes in the chart with circles. To create the straight lines between the circles, you could delete the elbow connector and then insert a straight connector.

Adding custom objects

Custom objects are objects that you create based on one of the existing objects from the **Palette** view.

About this task

To create a custom object, you select one of the objects from the **Palette** view and modify its properties to meet your needs. You can then store the custom object that you created on the **Custom** palette where it becomes available for repeatable use.

The **Custom** palette is particularly useful for storing objects that are used in every scene in your dashboard or on every page of your report. For example, reusing a title object that has the font, size, frame, color, and fill that you want. This saves you time and provides consistency throughout your project.

The **Custom** palette is a convenient place to store your company logo as a picture object. To do so, you would create a picture object, specify your logo image as the **ImageSource** property for the **Picture** object, and then store the customized **Picture** object to the **Custom** palette.

Note: You can only store object properties with a custom object. Events and parameters associated with an object are not stored and cannot be reused from instance to instance. You must manually re-assign event actions and parameterized properties to objects if you want to duplicate those attributes.

To create a custom object and add it to the **Custom** palette:

Procedure

1. In the **Palette** view, select the object that you want to customize. For example, to create a custom text object, locate the default **Label** object in the palette and insert it in the editor window.
2. From the **Properties** view, modify the properties of the object as you want.
3. When you are finished, return to the editor window, right-click the object and select **Add to Custom Palette** from the pop-up menu. The object retains the properties that you set and is placed on the **Custom** palette.

Results

Note:

- You can rename the objects on the **Custom** palette. To do so, right-click an object on the **Custom** palette and select **Customize**. The **Customize Palette** window opens. Enter the new name in the **Name** field and click **OK**.
- You can always change the default properties assigned to any customized object.
- The **Custom** palette can provide a single source for all the objects that you will use in your visual project. You can create several custom objects and place them on the **Custom** palette. In addition, you can place objects from any of the palettes in the **Palette** view on the **Custom** palette. That way all the objects that you will use in your project are located in one palette.

For example, if you use a Text object often in a dashboard but want the size to be 9 point rather than 12 point, and the font to be Palatino rather than Arial, you can save your customized Text object to the **Custom** palette. Perhaps you want to reuse a Marker object that is defined as a solid triangle shape, red color, and 9 point size; you can save the customized Marker object to the **Custom** palette.

Adding container objects

The **Containers** palette includes the **Embedded Scene**, **Embedded Content**, and **Canvas** objects.

About this task

Embedded Content

An object that provides the means to add external content to a visual report or dashboard. You view the embedded content in the embedded content object.

Embedded Scene

An object that provides the means to navigate from one scene to another. The embedded scene actually displays in the embedded scene object.

Canvas

An object that contains the visual dashboard data objects that you can manipulate in **Runtime** mode.

Inserting an embedded scene object

An **Embedded Scene** object is a container through which you can view all or a portion of another scene and then navigate to that scene.

About this task

You can use the **Embedded Scene** object only in visual dashboards, they are not applicable to visual reports. You can embed a scene if you have already defined two scenes. The source scene is where you insert the embedded scene object. The scene specified within the embedded scene object is considered the destination scene.

To insert an embedded scene object in your visual dashboard scene:

Procedure

1. From the **Project Explorer** view, open the source scene in the **Canvas** mode.
2. From the **Palette** view, click the **Embedded Scene** object.
3. Click in the editor window to insert the **Embedded Scene** object. The **Embedded scene** wizard opens.
4. Select the **Navigate to embedded scene when clicked with the mouse** check box to automatically generate an event action for the embedded scene object. The generated event action specifies that a mouse click the embedded scene object will take a user to the embedded scene. The **Navigate to embedded scene when clicked with the mouse** check box should be cleared if you want to specify a different event action for the embedded scene object.
5. If you want to navigate to a location from another visual project, select the **Another visual project** check box and click the ellipsis (...) to the right of the field.
The **Open** window opens.
6. Navigate to and select the visual project that you want the embedded scene to link to and click **Finish**.
The Open window closes and control returns to the **Embedded Scene Wizard**.
7. Select the name of the destination scene from the **Embedded scene** drop-down list.
8. Select the viewpoint of the embedded scene that you want to focus the window on in the **Viewpoints** list box.
9. Specify the coordinates and zoom level of the embedded scene's canvas in from the **Location** fields.
10. Click **OK**. An **Embedded Scene** object appears in the folder for your scene.

Inserting an embedded content object

An **Embedded Content** object is a container that provides the means to add external content to a visual report or dashboard. The external content that can be displayed can be a text file or a HTML page.

About this task

To insert an **Embedded Content** object in your visual dashboard scene:

Procedure

1. From the **Project Explorer** view, open the source scene in the **Canvas** mode.
2. From the **Palette** view, click the **Embedded Content** object.
3. Click in the editor window to insert the **Embedded Content** object. The **Embedded Content** object is drawn based on default size properties. Click in the editor window and drag the cursor to insert the **Embedded Content** object drawn to your specific size. The **Embedded content** wizard opens.
4. In the **Embedded Content mode** area click **ExternalData** to add a separate file or **DirectData** to add the content as a string.
5. If you selected **ExternalData** specify the location of the file you want to add in the **Content location** field. You can click the ellipsis (...) to search for the location.

6. Specify the type of content that will be embedded in the **Content type** field. Select **Text** to display content in text format and **HTML** to display content in HTML format.
7. If you selected **Text** content type, select the required character encoding from the **Character encoding** list.
8. If you selected **DirectData**, click the ellipsis (...) to add the required content in the **Content** field, and click **OK**.
9. Specify the type of content that will be embedded in the **Content type** field. Select **Text** to display content in text format and **HTML** to display content in HTML format.
10. If you selected **Text** content type, select the required character encoding from the **Character encoding** list.
11. Click **OK**. An **Embedded Content** object is created. An entry is made for the object in the **Project Explorer** view.

Inserting a Canvas object

A **Canvas** object contains the visual dashboard data objects that you can manipulate in **Runtime** mode.

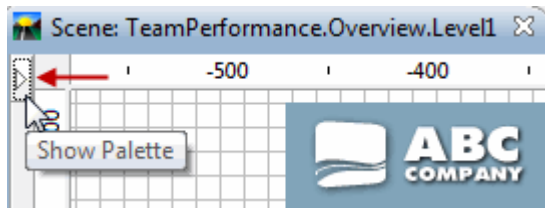
About this task

To insert a **Canvas** object in a visual dashboard:

Procedure

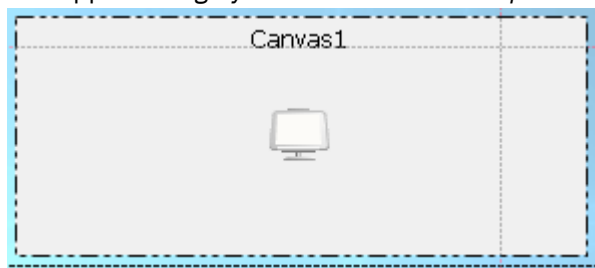
1. Make sure the visual dashboard is opened on the **Canvas** tab of the workbench editor area.
2. Click the **Show Palette** icon to show the list of folders that hold the elements that you can add to your visual dashboard.

Note: The **Show Palette** icon is located in the upper-right corner of the editor area:



3. From the **Containers** palette of the **Palette** view, select the **Canvas** object and insert it in the editor window.

The **Canvas** object is added to a visual dashboard. By default, the **Canvas** object is named **Canvas1** and appears as *grey box with a dotted line perimeter* in the editor area:



Results

The Canvas object has an internal tree. The repository subtree or queries subtree of data visualization objects and data transformation objects located in visual dashboard depends on the **Source** property of the internal tree.

The Canvas object has an internal tree that represents (in its entirety) the hierarchical structure of data visualization objects and data transformation objects. Keep in mind that Canvas supports only a subset of data visualization objects and data transformation objects.

To view the internal tree of the Canvas, switch the editor to **Runtime** mode.

Canvas types and layouts

Each Canvas type maintains a specific layout that depends on the objects that the Canvas contains. The layout of a Canvas is determined by the data objects placed on it and the size and location of the Canvas in the visual dashboard.

There are three types of Canvases:

Built-in

Built-in Canvas is stored inside the visual dashboard and is created while designing the dashboard with Canvas.

Built-in Canvas contains the objects and layouts that are the part of a visual dashboard. Typically, a Built-in Canvas represents what the designer believes to be the most user-appropriate set of data objects, as well as the layout of those objects.

The designer of the visual dashboard can also set the default type to a Built-in Canvas at design time. This default type is loaded for the users. If the designer of the visual dashboard does not set the default type, the users see the last canvas state that was used by the designer.

Users can adopt a Built-in Canvas and use it as a building block for their visual dashboards, tailoring it to suit the needs of their visual dashboard and then saving it under a new name. The tailored Canvas would be accessible, alongside those that come as part of the visual dashboard.

Shared

Shared Canvas is stored in the unified shared space in repository.

Shared Canvas is stored in a shared repository when the visual dashboard on which the canvas resides is open in **Runtime** mode. All repository users can access and use a Shared Canvas.

Personal

Personal Canvas is stored in the user home workspace.

Personal Canvas is accessed by users while the visual dashboard is in **Runtime** mode and stored in an existing personal workspace. If a personal workspace does not exist, users cannot save the canvas as a Personal Canvas. Each user can also set default type for current canvas that will be loaded for this user in the future.

Each object on the Canvas includes the following attributes:

- Key
- Container location and size
- Object layout (each object has to be able to save it's current layout).

Adding objects to the Canvas

You can add objects to a Canvas to provide functionality to your ad hoc report or visual dashboard.

About this task

To add objects to the Canvas of an ad hoc report, use the buttons on the Canvas toolbar.

Before adding objects to the Canvas objects of a visual dashboard, make sure:

- You created a visual dashboard and added a Canvas object to the dashboard.
- You placed the Canvas on the visual dashboard and sized the Canvas appropriately.
- You set the Canvas properties.

To add objects to the Canvas of a visual dashboard, switch the editor in the **Runtime** mode and use the buttons on the Canvas toolbar.

Adding objects to the Canvas in the HTML5 mode

You can add objects to the Canvas object of an ad hoc report or a visual dashboard in the HTML5 mode to provide functionality to the Canvas object.


About this task

To add objects to the Canvas, complete the following steps:

Procedure

1. If you work with a visual dashboard, click the **Runtime** tab in the editor to place the visual dashboard in the **Runtime** mode.

If you work with an ad hoc report, your Canvas is already in the **Runtime** mode.

2. Click  to expand the Canvas toolbar.
3. Click **Add from Repository** to open the Canvas internal tree of objects.
4. Expand the nodes of the tree to locate the objects that you want to add to your Canvas. Click **Add**.

Each object that you add to the Canvas can be displayed as a grid or a chart. Currently, Canvas supports the following objects:

- Analytical Query
- Compound Dynamart
- Drill-down Path
- Dynamart
- Table
- Visual Query

Note: You can add queries with prompts to the Canvas object of your ad hoc report or visual dashboard. For visual dashboards, use local or global parameters to manage prompts automatically. Insert a parameter to the visual project, make sure that it has the same name as one of the prompted variables used in the query, and set the default value. This value is set for the prompt variable when you add the query to the Canvas.

Also, for ad hoc reports and visual dashboards, you can specify prompt values when adding query to the Canvas.

5. You can add control objects (text box, combo box, and list box) and labels to your Canvas. Use the control objects to filter the data on your Canvas and to accept the user input.
 - To add the Label object, click **Add Label**. Use the Label object to create captions or explanatory notes to the data-driven objects (grids and charts) on your Canvas.
 - To add the Text box object, click **Add Text Box**. Use the Text box object, if you want the user to type the filter values manually.
 - To add the Combo box object, click **Add Combo Box**. Use the Combo box object, if you want to provide the user with a list of predefined filters for the data-driven objects. Combo box allows the user to select only one filter from the list.
 - To add the List box object, click **Add List Box**. Use the List box object, if you want to provide the user with a list of predefined filters for the data-driven objects. List box allows the user to select multiple filters from the list.

Results

You added some objects to the Canvas. Now, you can link these objects to filter the data that is displayed on your Canvas.


Editing objects properties in the HTML5 mode

You can customize the objects on the Canvas by editing the properties of these objects.

About this task

Each object on the Canvas has an individual toolbar. Use this toolbar to customize the object.

Procedure

1. Click the object to display the object toolbar.
2. The following buttons have identical or similar functions for all objects:
 - To change the position of the object on the Canvas, click  and drag the object to the new position.
 - To edit the title of the object, click **Edit Title**, type the new title, and specify the font options in the **Edit Title** window.
 - To see the source object path, click **About Source Object**.
 - To delete the object from the Canvas, click **Remove**.
3. To edit queries and charts, use the following buttons, in addition to the ones that are listed in step 2:
 - To change the display mode of your object, click **Display Mode** and select the new display mode.
 - To add a chart to the Canvas, click **Add Chart** and specify the settings of a new chart in the **Add New Chart** window.
 - To convert your current chart into a 2D chart, click **2D Charts** and select the chart type.
 - To convert your current chart into a 2D chart with depth, click **2D Charts with Depth** and select the chart type.
 - To change the source columns of a chart, click **Change Columns** and edit your chart in the **Chart Settings** window.
4. To edit the **Combo Box** and **List Box** objects, use the following buttons, in addition to the ones that are listed in step 2:
 - To populate your object with data from another object, click **Configure**. Select the source object (analytical query, visual query, chart, dynamart, compound dynamart, or table) from the **Resource** field and select the source object column that you want your combo box or list box to display from the **Column** list.
 - To define the dependency between the current object and another object on the Canvas, click **Create Wiring**. For more information, see the [Wiring objects on the Canvas in HTML5](#) topic.
 - To change the source object and source object column for your combo box or list box and to change the font settings, click **Change Settings**.
5. To edit the **Text Box** object, use the following buttons, in addition to the ones that are listed in step 2:
 - To define the dependency between the current object and another object on the Canvas, click **Create Wiring**. For more information, see the [Wiring objects on the Canvas in HTML5](#) topic.
 - To specify the default text for your text box object and to change the font settings, click **Change Settings**. The default text is displayed by your text box object when you open your ad hoc report or switch your visual dashboard into the **Runtime** mode.

Results

You customized the objects on the Canvas.

Managing Canvas operations using the Canvas toolbar

The Canvas toolbar provides actions for managing the Canvas in ad hoc reports and visual dashboards.

Canvas toolbar options

A use the Canvas toolbar to add an object to the Canvas:



Each option on the Canvas toolbar is described in the following table:

| Toolbar button | Description |
|----------------------------------|---|
| Create Layout | Click the Create Layout button to create an empty layout. |
| New Layout | Click the New Layout button to select layout from the list. Select Open default layout to switch to default Canvas layout. |
| Save Layout | Click the Save Layout button to save the current layout. |
| Save Layout As | Click the Save Layout as button to save the current layout as a Built-in, Shared (for dashboards only), or Personal Canvas (for dashboards only) type. |
| Set Layout as Default | Click the Set Layout as Default button to make the current layout default. |
| Clear Layout | Click the Clear Layout button to remove all objects from the current layout. |
| Export Layout | Click the Export Layout button to open the export dialog. In the dialog, specify the file name, the export area, and select the export format: PDF or PNG. |
| Toggle Wiring Lines | Click the Toggle Wiring Lines button to display or hide the wiring lines. |
| Switch to Normal Mode | Click the Switch to Normal Mode button to minimize the Canvas. |
| Switch to Fullscreen Mode | Click the Switch to Fullscreen Mode button to maximize the Canvas. |
| Add from Repository | Click the Add from Repository button to display the Canvas internal tree of objects. |
| Add Label | Click the Add Label button to add the label object to the Canvas. |
| Add Text Box | Click the Add Text Box button to add the Text Box object to the Canvas. |
| Add Combo Box | Click the Add Combo Box button to add the Combo Box object to the Canvas. |

Table 35. Toolbar options (continued)

| Toolbar button | Description |
|---------------------|--|
| Add List Box | Click the Add List Box button to add the List Box object to the Canvas. |

Wiring objects on the Canvas

Wiring objects defines the relationship and dependencies between objects. Use the object Wiring Mode to connect objects on the Canvas.

About this task

The following topics describe the process of wiring objects on the Canvas.

Wiring objects on the Canvas in HTML5

Use the object Wiring Mode to connect objects on the Canvas in HTML5.


About this task

Wiring objects defines the relationship and dependencies between objects. For example, you can link the DEMO.EMPLOYEE table with a combo box that contains the list of all position titles from that table. When you connect the two objects, the table only displays the employees who occupy the position, that is selected in the combo box.

To wire objects in HTML5, do the following actions:

Procedure



1. Click  to expand the Canvas toolbar.
2. Click **Create Wiring** of the first object and drag the wiring line to the second object.

Note: The **Create Wiring** buttons of objects that can be wired are highlighted in green. Any data-driven object that supports wiring can be the source object and the target object at the same time.

The **Create Link** window opens.

3. In the window, specify the source object column and the target object column.

If the queries already contain specified output and input columns, these columns are displayed and automatically selected. Otherwise, all the columns from the result set are displayed. For more information about input and output columns, see the [Managing input and output columns for the visual query](#) topic.

4. Click **Add**.
5. Optional: Click **Toggle Wiring Lines** on the Canvas toolbar to display or hide the wiring lines between the objects on the Canvas.

Results

You connected the objects.

Filtering objects on the Canvas

You can link a control object (combo box, list box, or text box) with a data-driven object to filter the information displayed by the data-driven object. This example is based on the Q.STAFF table from the Samples repository. In this example, we use the combo box control object for filtering.

Before you begin

Prepare two source queries:

- Create a visual query and type the following SQL statement for it: `select * from q.staff`. Name this query `Query 1` and save it to your workspace.
- Create another visual query and type the following SQL statement for it: `select distinct job from q.staff`. Name this query `Query Combo` and save it to your workspace.

About this task

You can use the combo box, list box, or text box object to filter your data-driven object. If you want to provide users with a list of predefined filters, use the combo box object or the list box object. If you want your users to type the filter text manually, use the text box object. In this example, we use the combo box object for filtering.

To link objects on the Canvas, perform the following procedure:

Procedure

1. Create an Ad hoc report.
2. Add the objects to your Ad hoc report.
 - a) In the Canvas toolbar, click **Add from Repository**, select `Query 1` from the tree, and click **Add**.
 - b) In the Canvas toolbar, click **Add Combo Box** to add a combo box. Click **Configure Combo Box** on the combo box toolbar and select `Query Combo` from the tree.
3. Link your combo box with your table.
 - a) Click **Create Wiring** on the combo box toolbar, drag the wiring line to your table, and click the table.
 - b) In the **Create Link** window, select **JOB** from both lists and click **Add**.

Results

Now, your combo box filters the information, that is displayed by your table.

Editing linked objects

Use the **Edit Link** function to make changes to the source and target objects or to remove the link relationship between the objects.

About this task

To edit linked objects, perform the following procedure:

Procedure

1. On the Canvas, click the source object.
The wiring line between the source object and the target object appears.
2. Double-click the wiring line.
The **Edit Link** window opens.
3. In the window, edit the source columns of the source object and destination columns of the target object in the corresponding fields.
4. To remove the link between the two objects, click **Delete**.
The link between the objects is removed and the **Edit Link** window closes.
5. To save the changes and close the **Edit the Link** window, click **Save**.
6. To discard the changes and close the **Edit the Link** window, click **Cancel**.

Managing input and output columns for the visual query

You can specify available input and output columns for the current visual query. These columns will be used as source columns and target columns for wiring objects on a canvas when creating visual dashboards or ad hoc reports.

About this task

The steps in this procedure describe how to specify input and output columns for a query.

Procedure

1. Select **Query > Wiring Settings** to open the **Wiring Settings** window.
2. In the **Wiring Settings** window the list of available columns is displayed. To specify input and output columns select the corresponding check boxes.
3. After you have specified the input and output columns, click **OK** to apply the changes and close the **Wiring Settings** window.

Working with layout objects

The **Layouts** palette contains graphical objects that you can use to display the data that has been obtained as a result of a query. Layout objects include a wide assortment of charts, graphs, maps, tables and grids.

About this task

The following chart layouts are available:

- Bar Chart
- Pie Chart
- Column Chart
- Scatter Chart
- XY Chart
- Stock Chart
- Candlestick Chart
- Timeline Chart
- Event Band Chart
- Multivariate Chart
- Linear Map

The following hierarchy layouts are available for showing dependency relationships:

- Cluster Graph
- Organization Chart
- Tree Chart

The following pattern layouts are available for showing data in an ordered pattern but without quantity or dependency relationships:

- Table
- Grid
- Simple Form
- Matrix
- Spiral
- Horizon

The Solid Chart object is a monolithic object. It can be used to display your query results data in a visually rich format, like charts for queries.

Inserting a table

Table objects let you display query results in a simple table.

About this task

To insert a Table object:

Procedure

1. Double-click the **Table** object in the **Layouts** palette of the **Palette** view.
The **Table Wizard** opens.
2. Specify whether you want to populate the table with data from an existing query and click **Next**.
3. If you specified **No**, specify the table options and click **Finish**.
The wizard closes and the table is displayed in the editor window.
4. If you specified **Yes**, select the query that you want to use to populate the table and click **Next**.
5. Select the columns that you want to include in the table and click **Next**.
6. Specify the table options and click **Next**.
7. Specify paging options and click **Finish**.
The wizard closes and the table is displayed in the editor window.

Inserting a simple table

You can insert a simple table object that is not populated with data from an existing query.

About this task

To insert a simple Table object:

Procedure

1. Double-click the **Table** object in the **Layouts** palette of the **Palette** view.
The **Table Wizard** opens.
2. Select **No** and click **Next**.
The **Specify table options** page opens.
3. Specify the number of rows the table will have in the **Rows** list.
4. Specify the number of columns the table will have in the **Columns** list.
5. Specify the alignment of the text in the cells in the **Alignment** drop-down list.
The valid options are:
 - Left
 - Right
 - Center
6. Specify the minimum width of each cell in the **Minimum width** field.
7. Click **Finish**.
The wizard closes and the table is displayed in the editor window.

Inserting a populated table

You can insert a table object that is populated with data from an existing query.

About this task

To insert a populated Table object:

Procedure

1. Double-click the **Table** object in the **Layouts** palette of the **Palette** view.
The **Table Wizard** opens.
2. Select **Yes** and click **Next**.
The **Select the query** page opens.
3. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
4. Select one or more columns in the **Available fields** list box and click **Move Right** to move them to the **Columns to be created** list box.
This field lists the names of the columns of data that are included in the selected query result set.
The **Specify table options** page opens.
5. Select the **Include title row** check box to create a title row.
6. Select the **Include header row** check box to create a header row.
7. Select the **Include footer row** check box to create a footer row.
8. Select the **Word-wrap cell text** check box to make the text in each cell fit the cell space.
9. Specify the alignment of the text in the cells in the **Alignment** list.
The valid options are:
 - Left
 - Right
 - Center
10. Specify the minimum width of each cell in the **Minimum width** field.
11. Specify the height limit of the table in the **Height Limit** field. Zero (0) value means that the table height is unlimited. Any value different from zero (0) sets the maximum table height.
12. Select the necessary **Scroll Scope** value from the list:
 - **Entire Table** to scroll the whole table.
 - **Data Only** to scroll only table data.
 - **Data and Heading** to scroll only table data and column headers.
13. Select the necessary value from **Vertical Scroll** list to set the vertical scrolling.
14. Specify the width limit of the table in the **Width Limit** field. Zero (0) value means that the table width is unlimited. Any value different from zero (0) sets the maximum table width.
15. Click **Next**.
The **Specify paging options** page opens.
16. Select the **Enable paging** check box to enable paging.
Paging allows you to display only a certain amount of rows at a time.
17. Specify the amount of rows that you want to display at a time in the **Rows per page** list.
18. Select the first row in the result set to display in the **First displayed row** list.
19. Select the **Add navigation bar** check box to automatically create page forward and page backwards buttons that control paging actions.
20. Click **Finish**.
The wizard closes and the table is displayed in the editor window.

Including a summary row

You can include a summary row in a table using the **Show Total Summaries** menu item.

About this task

By selecting the **Show Total Summaries** command in the associated query's **Results** menu, you can include a summary row with aggregated totals in the table object. To include a summary row in a table:

Procedure

1. Insert a table object that is populated with data from an existing query into the dashboard canvas.
2. Open the **Queries** folder in the **Project Explorer** and double-click the table object's associated query.
The query opens in the query editor.
3. Run the query.
The query results are displayed in the query results editor.
4. Select **Results > Show Total Summaries**.
A summary row is added to the query results grid.
5. Right-click a column header label, select **Grouping and Aggregation**, and select the aggregated total that you want to add to the summary row.
Repeat this for any column that you want to add an aggregated total to.
6. Save and close the query.
7. Run the visual dashboard.
The summary row is displayed, with the aggregated totals that you selected, in the table object's grid.

Formatting a summary row

By using the **isSummaryRow()** function, you can add formatting to a table object's summary row.

About this task

The **isSummaryRow()** expression designer function checks whether a row in a table object's grid is a normal row of the result set, or the summary row of the result set. If the row is the summary row, it returns true. Using this functionality, you can add specific formatting to a table object's summary row. To add formatting to a table object's summary row:

Procedure

1. In the **Project Explorer**, navigate to the table object's data template.
2. To exclude the summary row, select the table row in the data template and click the **Value** field of the **Visible** property in the **Properties** view.
3. Click the **Edit with Expression Designer** button on the **Properties** view toolbar.
The **Expression Designer** opens.
4. Delete the text in the field and double-click the **if** function in the **Logical** function list.
`=if(bConditional, vIfTrue, vIfFalse)` is inserted into the field with `bConditional` highlighted.
5. Double-click the **isSummaryRow** function in the **Information** function list.
`isSummaryRow()` is inserted into the function.
6. Replace `vIfTrue` with `false` and `vIfFalse` with `true`.
7. Click **OK**.
The **Expression Designer** closes.
8. To add formatting to the summary row's cells or content, select the table cell or table cell label in the data template and click the **Value** field of the property that you want to format in the **Properties** view.
9. Click the **Edit with Expression Designer** button on the **Properties** view toolbar.
The **Expression Designer** opens.
10. Delete the text in the field and double-click the **if** function in the **Logical** function list.
`=if(bConditional, vIfTrue, vIfFalse)` is inserted into the field with `bConditional` highlighted.
11. Double-click the **isSummaryRow** function in the **Information** function list.
`isSummaryRow()` is inserted into the function.

12. Replace `vIfTrue` with the formatting that you want to apply to the summary row of the query result set. Replace `vIfFalse` with the formatting that you want to apply to the body rows of the query result set.
13. Click **OK**.
The **Expression Designer** closes.

Inserting a grid

The **Grid** object displays query results in a simple grid.

About this task

To insert a grid:

Procedure

1. Double-click the **Grid** object in the **Layouts** palette of the **Palette** view.
2. In the **Grid Wizard**, navigate to the table or query that will provide data for the **Grid** object.
3. Click **Finish**.

The **Grid Wizard** closes and the **Grid** object is displayed on the scene.

Inserting a bar or column chart

Bar charts and column charts display data as a series of bars on an x or y axis.

About this task

To insert a bar or column chart:

Procedure

1. Double-click the **Bar Chart** or **Column Chart** object in the **Layouts** palette of the **Palette** view.
The **Bar Chart Wizard** or **Column Chart Wizard** opens.
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The **Select layout type** page opens.
3. Click **Standard** and click **Next**.
The **Select the field(s) to be displayed as bars** page opens.
4. Select one or more columns in the **Available fields** list box and click **Move Right** to move them to the **Fields to be displayed** list box.
This field lists the names of the columns of data that are included in the selected query result set. The columns that you move to the **Fields to be displayed** list box will be displayed as the bars along a labeled axis in the column or bar chart.
5. Click **Next**.
The **Set up labels and legend information for chart** page opens.
6. Select the **Include a label** check box to include a label that displays for each value bar in the chart.
7. Specify a data column in the **Label field** drop-down list that will be used to populate the labels for each value bar in the chart.
This field is only valid if you selected the **Include a label** check box.
For example, if you select a column listing salaries in the **Fields to be displayed** list box, and a column listing the names of employees in the **Label field** drop-down list, the chart will display the name of each person next to value bars that show their individual salaries.
8. Select the **Display the value of each bar** check box to display the data value at the end of each value bar.

9. Select the **Stack bars when there are multiple series** check box to stack value bars of different data series on the same line.

The value bars representing the first data column in the **Fields to be displayed** list box are drawn first, and subsequent value bars start where the ones before them end.

For example, if bar 1 has a value of 50 and bar 2 has a value of 20, bar 1 is displayed from 0 to 50 and bar 2 is displayed from 50 to 70.

10. Click **Next**. The **Specify paging options** page opens.
11. Select the **Enable paging** check box to enable paging.

Paging allows you to display only a certain amount of rows at a time. For instance, if you only display ten rows at a time, only ten columns or bars will be displayed for each page.
12. Select the amount of rows that you want to display at a time in the **Rows per page** list.
13. Select the first row in the result set to display in the **First displayed row** list.
14. Select the **Add navigation bar** check box to automatically create page forward and page backwards buttons that control paging actions.
15. Click **Finish**.

The wizard closes, and the bar or column chart is displayed in the editor window.

Inserting an extended bar or column chart

Extended bar and column charts allow you to roll up data across one or more columns in a result set.

About this task

To insert an extended bar or column chart:

Procedure

1. Double-click the **Bar Chart** or **Column Chart** object in the **Layouts** palette of the **Palette** view.

The **Bar Chart Wizard** or **Column Chart Wizard** opens.
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.

The **Select layout type** page opens.
3. Click **Extended** and click **Next**.

The **Select the field(s) to be displayed as bars** page opens.
4. Select one or more columns in the **Available fields** list box and click **Move Right** to move them to the **Field** column of the **Fields to be displayed** table.

This field lists the names of the columns of data that are included in the selected query result set. The columns that you move to the **Fields to be displayed** table will be displayed as the bars along a labeled axis in the column or bar chart.
5. Click a cell in the **Aggregation** column of the **Fields to be displayed** table. A drop-down list opens. Choose an option to specify how the column is aggregated.
6. Click **Next**.

The **Select one or more fields to be moved in the side group** page opens.
7. Select one or more columns in the **Available fields** list box and click **Move Right** to move them to the **Fields to be moved** list box.

The columns that you move to the **Fields to be moved** list box will be rolled up in the query result set, and the order of the columns will determine the roll up hierarchy.
8. Click **Next**.

The **Set up labels and legend information for chart** page opens.
9. Select the **Display the value of each bar** check box to display the data value at the end of each value bar.
10. Select the **Stack bars when there are multiple series** check box to stack value bars of different data series on the same line.

The value bars representing the first data column in the **Fields to be displayed** list box are drawn first, and subsequent value bars start where the ones before them end.

For example, if bar 1 has a value of 50 and bar 2 has a value of 20, bar 1 is displayed from 0 to 50 and bar 2 is displayed from 50 to 70.

11. Click **Finish**.

The wizard closes, and the bar or column chart is displayed in the editor window.

Displaying OLAP data in a bar or column chart

You can display OLAP data in a bar or column chart by using a multidimensional query to obtain query results.

About this task

Bar or column charts that are run using multidimensional queries allow you to roll up data across one or more columns in the associated query result set. This allows users to drill down into specific data symbols to obtain only the data they need, without having to view the entire result set at once.

To insert a bar or column chart that displays OLAP data:

Procedure

1. Double-click the **Bar Chart** or **Column Chart** object in the **Layouts** palette of the **Palette** view.

The **Bar Chart Wizard** or **Column Chart Wizard** opens.

2. In the **All Data Sources** tree, navigate to an OLAP query in the **Multidimensional Data Sources** folder that will be run to obtain the layout object's query result set and click **Next**.

The **Select the field(s) to be displayed as bars** page opens.

3. Select one or more columns in the **Available fields** list box and click the **Move Right** button to move them to the **Fields to be displayed** list box.

This field lists the names of the columns of data that are included in the selected query result set. The columns that you move to the **Fields to be displayed** list box will be displayed as the bars along a labeled axis in the column or bar chart.

4. Click **Next**.

The **Select dimension or hierarchy** page opens.

5. Select one or more dimensions of the cube model that you want to use to express the query results and click **Next**.

Select more than one dimension by holding the **Ctrl** key and clicking the dimensions that you would like to use.

The **Set up labels and legend information for chart** page opens.

6. Select the **Display the value of each bar** check box to display the data value at the end of each value bar.

7. Select the **Stack bars when there are multiple series** check box to stack value bars of different data series on the same line.

The value bars representing the first data column in the **Fields to be displayed** list box are drawn first, and subsequent value bars start where the ones before them end.

For example, if bar 1 has a value of 50 and bar 2 has a value of 20, bar 1 is displayed from 0 to 50 and bar 2 is displayed from 50 to 70.

8. Click **Finish**.

The wizard closes, and the bar or column chart is displayed in the editor window.

Inserting a pie chart using a relational query

Pie charts display data in a circular chart that is cut into wedges.

About this task

To insert a pie chart using a relational query:

Procedure

1. Double-click the **Pie Chart** object in the **Layouts** palette of the **Palette** view.
The **Pie Chart Wizard** opens.
2. In the **All Data Sources** tree, navigate to the table or relational query that will be run to obtain the layout object's query result set and click **Next**.
The **Select layout type** page opens.
3. Click **Standard** and click **Next**.
The **Select options for the pie chart** page opens.
4. In the **Wedge values** drop-down list, specify the name of the column of data in your query result set that will contain the values for each pie wedge in the pie chart.
You must select a data column from the drop-down list of available data columns in your query result set.
5. Select the **Include a label for the wedge** check box to include a label for each wedge.
6. In the **Label field** drop-down list, specify the data column in your query result set that will contain the label information for each pie wedge.
You must select a field from the drop-down list of available data columns from you query result set if you selected the **Include a label for the wedge** check box.
7. Select the **Include the value of the wedge** check box to display in the pie chart the value of each wedge.
8. Select the **Include the percentage the value represents of the whole pie** check box to display a percentage value for each pie wedge.
9. Click **Finish**.
The **Pie Chart Wizard** closes and the chart is displayed in the editor window.

Inserting an extended pie chart

Extended pie charts allow you to roll up data across one or more columns in a result set.

About this task

To insert an extended pie chart:

Procedure

1. Double-click the **Pie Chart** object in the **Layouts** palette of the **Palette** view.
The **Pie Chart Wizard** opens.
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The **Select layout type** page opens.
3. Click **Extended** and click **Next**.
The **Select one or more fields to be moved in the side group** page opens.
4. Select one or more columns in the **Available fields** list box and click **Move Right** to move them to the **Fields to be moved** list box.
The columns that you move to the **Fields to be moved** list box will be rolled up in the query result set, and the order of the columns will determine the roll up hierarchy.
5. Click **Next**.

The **Select options for the pie chart** page opens.

6. In the **Wedge values** drop-down list, specify the name of the column of data in your query result set that will contain the values for each pie wedge in the pie chart.
You must select a data column from the drop-down list of available data columns in your query result set.
7. In the **Aggregation** drop-down list, specify how the column of data specified in the **Wedge values** field will be aggregated.
8. Select the **Include a label for the wedge** check box to include a label for each wedge.
9. In the **Label field** drop-down list, specify the data column in your query result set that will contain the label information for each pie wedge.
You must select a field from the drop-down list of available data columns from you query result set if you selected the **Include a label for the wedge** check box.
10. Select the **Include the value of the wedge** check box to display in the pie chart the value of each wedge.
11. Select the **Include the percentage the value represents of the whole pie** check box to display a percentage value for each pie wedge.
12. Click **Finish**.

The **Pie Chart Wizard** closes and the chart is displayed in the editor window.

Displaying OLAP data in a pie chart

You can display OLAP data in a pie chart by using a multidimensional query to obtain query results.

About this task

Pie charts that are run using multidimensional queries allow you to roll up data across one or more columns in the associated query result set. This allows users to drill down into specific data symbols to obtain only the data they need, without having to view the entire result set at once.

To insert a pie chart that displays OLAP data:

Procedure

1. Double-click the **Pie Chart** object in the **Layouts** palette of the **Palette** view.
The **Pie Chart Wizard** opens.
2. In the **All Data Sources** tree, navigate to an OLAP query in the **Multidimensional Data Sources** folder that will be run to obtain the layout object's query result set and click **Next**.
The **Select dimension or hierarchy** page opens.
3. Select one or more dimensions of the cube model that you want to use to express the query results and click **Next**.
Select more than one dimension by holding the **Ctrl** key and clicking the dimensions that you would like to use.
The **Select options for the pie chart** page opens.
4. In the **Wedge values** drop-down list, specify the name of the column of data in your query result set that will contain the values for each pie wedge in the pie chart.
You must select a data column from the drop-down list of available data columns in your query result set.
5. Select the **Include a label for the wedge** check box to include a label for each wedge.
6. In the **Label field** drop-down list, specify the data column in your query result set that will contain the label information for each pie wedge.
You must select a field from the drop-down list of available data columns from you query result set if you selected the **Include a label for the wedge** check box.
7. Select the **Include the value of the wedge** check box to display in the pie chart the value of each wedge.

8. Select the **Include the percentage the value represents of the whole pie** check box to display a percentage value for each pie wedge.
9. Click **Finish**.

The **Pie Chart Wizard** closes and the chart is displayed in the editor window.

Inserting a multivariate chart

Multivariate charts display data in a grid of scatter charts.

About this task

To insert a multivariate chart:

Procedure

1. Double-click the **Multivariate Chart** object in the **Layouts** palette of the **Palette** view.
The **Multivariate Chart Wizard** opens.
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The **Select the fields to be mapped to axes** page opens.
3. Select one or more columns in the **Available fields** list box and click the **Move Right** button to move them to the **Axes to be created** list box.
This field lists the columns that have been selected from your query result set whose data will be mapped to an axis in the multivariate chart. The axes of the multivariate chart are labeled along the diagonal, with the first axis drawn in the lower left grid cell using the first column of data specified in the **Axes to be created** list box. Each column of data is charted along the diagonal in the order that it appears in the **Axes to be created** list with the last axis drawn in the upper right grid cell.
4. Click **Next**.
The **Specify multivariate chart options** page opens.
5. Specify the width of each cell in the **Width** field.
6. Specify the Height of each cell in the **Height** field.
7. Specify the gutter width (space between cells horizontally) in the **Horizontal gutter** field.
8. Specify the gutter height (space between cells vertically) in the **Vertical gutter** field.
9. Click **Finish**.
The **Multivariate Chart Wizard** closes and the chart is displayed in the editor window.

Inserting a scatter or XY chart

Scatter charts display data in a two-dimensional graph in rectangular coordinates. XY charts display data in a two-dimensional graph consisting of connected points whose coordinates represent two variables.

About this task

To insert a scatter or XY chart:

Procedure

1. Double-click the **Scatter Chart** or **XY Chart** object in the **Layouts** palette of the **Palette** view.
The **Scatter Chart Wizard** or **XY Chart Wizard** opens.
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The **Select the field(s) to be displayed as Y values** page opens.
3. Select one or more columns in the **Available fields** list box and click the **Move Right** button to move them to the **Fields to be displayed** list box.

This field lists the names of the columns of data that are included in the selected query result set. The columns that you move to the **Fields to be displayed** list box will be displayed as Y values in the generated chart.

4. Click **Next**.

The **Select the fields representing the point location** page opens.

5. In the **X value** drop-down list, specify the name of the column of data from your query result set that will be graphed on the x-axis of the scatter or XY chart.

Select a data column field from the drop-down list of available data columns in your query result set.

6. Click **Finish**.

The wizard closes and the chart is displayed in the editor window.

Inserting a stock chart

Stock charts are used to show fluctuations in stock prices, or other value-based indices, and volumes over time.

About this task

To insert a stock chart:

Procedure

1. Double-click the **Stock Chart** object in the **Layouts** palette of the **Palette** view.

The **Stock Chart Wizard** opens.

2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.

The **Select options for a stock chart** page opens.

3. In the **Trading interval** drop-down list, select the trading interval over which the stock values are to be charted.

Select an interval from the drop-down list of available choices that includes a second, minute, hour, day, week, month or year.

4. In the **Date** drop-down list, specify the name of the data column in your query result set that contains the trading date.

5. In the **High** drop-down list, specify the name of the data column in your query result set that will contain the stock's high price on each date.

6. In the **Low** drop-down list, specify the name of the data column in your query result set that will contain the stock's low price on each date.

7. Select the **Show open price** check box to display the open price for the stock.

8. In the **Open** drop-down list, select the name of the data column in your query result set that will contain the stock's open price on each date.

9. Select the **Show close price** check box to display the close price for the stock.

10. In the **Close** drop-down list, select the name of the data column in your query result set that will contain the stock's close price on each date.

11. Select the **Show volume of trading** check box to display the volume of trading for each stock.

12. In the **Volume** drop-down list, select the name of the data column in your query result set that will contain the volume that the stock traded on each date.

13. Click **Finish**.

The **Stock Chart Wizard** closes and the chart is displayed in the editor window.

Inserting a candlestick chart

Candlestick charts are used to show gains and losses in stock prices (or other value-based indices) and volumes over time.

About this task

To insert a candlestick chart:

Procedure

1. Double-click the **Candlestick Chart** object in the **Layouts** palette of the **Palette** view.
The **Candlestick Chart Wizard** opens.
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The **Select options for a candlestick chart** page opens.
3. In the **Trading interval** drop-down list, select the trading interval over which the stock values are to be charted.
Select an interval from the drop-down list of available choices that includes a second, minute, hour, day, week, month or year.
4. In the **Date** drop-down list, select the name of the data column in your query result set that contains the trading date.
5. In the **Open** drop-down list, select the name of the data column in your query result set that will contain the stock's open price on each date.
6. In the **Close** drop-down list, select the name of the data column in your query result set that will contain the stock's close price on each date.
7. Select the **Show high price** check box to display the high price for the stock.
8. In the **High** drop-down list, select the name of the data column in your query result set that will contain the stocks high price on each date.
9. Select the **Show low price** check box to display the low price for the stock.
10. In the **Low** drop-down list, select the name of the data column in your query result set that will contain the stock's low price on each date.
11. Select the **Show volume of trading** check box to display the volume of trading for each stock.
12. In the **Volume** drop-down list, select the name of the data column in your query result set that will contain the volume that the stock traded on each date.
13. Click **Finish**.
The **Candlestick Chart Wizard** closes and the chart is displayed in the editor window.

Inserting a timeline chart

Timeline charts display a time series as a curve.

About this task

To insert a timeline chart:

Procedure

1. Double-click the **Timeline Chart** object in the **Layouts** palette of the **Palette** view.
The **Timeline Chart Wizard** opens.
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The Select the fields representing the point location page opens.

3. In the **Date** drop-down list, specify the name of the data column from your query result set that will be graphed on the date/time axis of the timeline chart.
You should select a data column that includes a date or time value
4. In the **Value** drop-down list, specify the name of the data column from your query result set that will be graphed on the numeric axis of the timeline chart.
You should select a data column that includes numeric values.
5. Click **Finish**.
The **Timeline Chart Wizard** closes and the chart is displayed in the editor window.

Inserting an event band chart

Event band charts display a vertical band that represents time intervals for each row returned by a query.

About this task

To Insert an event band chart:

Procedure

1. Double-click the **Event Band Chart** object in the **Layouts** palette of the **Palette** view.
The **Event Band Chart Wizard** opens.
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The **Select the fields representing the event time period** page opens.
3. In the **Begin date** drop-down list, specify the name of the data column from your query result set that contains the beginning date of the event you are charting in the Event Band chart.
You should select a data column that includes a date.
4. In the **End date** drop-down list, specify the name of the data column from your query result set that contains the end date of the event you are charting in the Event Band chart.
You should select a data column that includes a date.
5. Click **Finish**.
The **Event Band Chart Wizard** closes and the chart is displayed in the editor window.

Inserting a cluster graph

Cluster graphs display data in a recursive and circular set of hubs with spokes.

About this task

To insert a cluster graph:

Procedure

1. Double-click the **Cluster Graph** object in the **Layouts** palette of the **Palette** view.
The **Cluster Graph Wizard** opens.
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The **Select the fields representing the parent-child relationship** page opens.
3. From the **Parent field** drop-down list, select the name of the data column in your query result set that will be used for the parent cell in the Cluster graph.
4. From the **Child field** drop-down list, select the name of the data column in your query result set that will be used as the child cell in the Cluster graph.
5. In the **Select the maximum visible child depth** list box, specify the maximum number of child levels that are displayed in the cluster graph under the parent objects for a given zoom level.

To see a single level of child levels, type a 1 in the field.

6. Click **Finish**.

The **Cluster Graph Wizard** closes and the graph is displayed in the editor window.

Inserting a linear map

Linear maps display spatial data in a geographic context.

About this task

To insert a linear map:

Procedure

1. Double-click the **Linear Map** object in the **Layouts** palette of the **Palette** view.
The **Linear Map Wizard** opens.
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The **Linear map options** page opens.
3. Select **Data contains open boundaries. Plot as Polylines.** to specify that your linear map will be plotted as polylines.
Choose this option if the vertices data is stored in your query result data columns as polylines. When stored as polylines, the vertices have open boundaries. The last point and the first point are different.
4. Select **Data contains closed boundaries. Plot as polygons.** to specify that your linear map will be plotted as polygons.
Choose this option if the vertices data is stored in your query result data columns as polygons. When stored as polygons, the vertices have closed boundaries.
5. Click **Next**.
The **Select the field containing the boundaries** page opens.
6. In the **Boundary Field** drop-down list, specify the data column in your query result set that contains the map vertices.
7. Click **Finish**.
The **Linear Map Wizard** closes, and the map displays in the editor window.

Inserting a Simple Form object

The **Simple Form** object allows you to optimize data retrieving process when you need to display values from only one row in a query.

About this task

You can put several **Label** objects on a **Simple Form** to display values from different cells of the specified query row.

To insert a Simple Form:

Procedure

1. Double-click the **Simple Form** object in the **Layouts** palette of the **Palette** view.
2. In the **Simple Form Wizard**, navigate to the table or query that will provide data for the **Simple Form** layout and click **Finish**.
The **Simple Form Wizard** closes and the **Simple Form** object is placed on the scene.
3. Add required **Label** objects to the **Simple Form** layout.
 - a) In the **Project Explorer** view, navigate to the **Simple Form** layout that you inserted.
 - b) In the **Palette** view, expand **Primitives** and select **Label**.

- c) Drag the **Label** object to the **Level1** node under **Simple Form** layout in the **Project Explorer** view.
4. With the help of **Expression Designer**, specify the value to display in each **Label** object.
 - a) In the **Project Explorer** view, select the **Label** object you added.
 - b) In the **Properties** view, double-click the **Text** property value.
 - c) In the **Expression Designer** window, replace default text with the name of the column that contains the value you want to display. For this, expand **Queries > [Query name] > Output columns** and select the required column.
Note: [Query name] stands for the name of the query or table that you specified for the **Simple Form** layout.
5. Click **OK** to close the **Expression Designer** window.

Results

The **Simple Form** layout with all required labels is added to the scene. You can check the result in Runtime mode.

Inserting a matrix

Matrices display data in a two-dimensional array of cells.

About this task

To insert a matrix:

Procedure

1. Double-click the created object and add any controls (for example, labels) with the necessary values to display them in the [object_name].
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The **Specify matrix options** page opens.
3. Specify the width of each cell in the **Cell width** field.
4. Specify the Height of each cell in the **Cell height** field.
5. Specify the gutter width (space between cells horizontally) in the **Gutter width** field.
6. Specify the gutter height (space between cells vertically) in the **Gutter height** field.
7. Specify the number of columns that will be displayed in the matrix in the **Number of columns** field.
8. Select the **Display cell border** check box to display a border around each cell in the matrix.
9. Click **Finish**.
The **Matrix Wizard** closes and the matrix is displayed in the editor window.

Inserting a spiral

The **Spiral** object displays data in an infinite inward spiral, in which each data point becomes smaller.

About this task

The Spiral layout is useful for displaying data that is not mathematically related. Zooming in on the **Spiral** object reveals more data further into the spiral, and zooming out shows more data on the outer portion of the spiral.

To insert a spiral:

Procedure

1. Double-click the **Spiral** object in the **Layouts** palette of the **Palette** view.
2. In the **Spiral Wizard**, navigate to the table or query that will provide data for the **Spiral** object.

3. Click **Finish**.

The **Spiral Wizard** closes and the **Spiral** object is displayed on the scene.

Results

You can customize the look and behavior of a Spiral object by modifying its properties and properties of its supporting objects: DataTemplate and Ellipse.

Working with nodes of Tree Chart and Organization Chart

You can insert **Tree Chart** or **Organization Chart** into a visual dashboard and expand or collapse its nodes.

Inserting an organization chart

Organization charts display data in a tree structure that reads from top to bottom and is typically associated with a personal hierarchy.

About this task

To insert an organization chart:

Procedure

1. Double-click the created object and add any controls (for example, labels) with the necessary values to display them in the [object_name].
2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The **Organization chart options** page opens.
3. In the **Parent field** drop-down list, select the name of the data column in your query result set that will be used for the parent cell in the Organization chart.
4. In the **Child field** drop-down list, select the name of the data column in your query result set that will be used as the child cell in the Organization chart.
5. Select the **Include a label** check box to include a label in each of the parent and child cells of the Organization chart.
6. In the **Label field** drop-down list, specify the name of a data column in your query result set that will contain the label information for each parent and child cell in the organization chart.
You must select a data column from the drop-down list of available data columns that are in your query result set if you selected the **Include a label** check box.
7. In the **Size of child objects relative to parent objects** field, specify the size of the child objects as they are displayed in the Organization chart relative to the size of the parent objects.
Type a number between .01 and 1 in the field.
8. In the **Layout Model** drop-down list, select the layout view option either classic or horizontal.
9. Click **Finish**.
The **Organization Chart Wizard** closes and the chart is displayed in the editor window.

Inserting a tree chart

Tree charts display data in a tree structure showing a cascading set of parent-child relationships.

About this task

To insert a tree chart:

Procedure

1. Double-click the created object and add any controls (for example, labels) with the necessary values to display them in the [object_name].

2. In the **All Data Sources** tree, navigate to the table or query that will be run to obtain the layout object's query result set and click **Next**.
The **Tree chart options** page opens.
3. In the **Parent field** drop-down list, select the name of the data column in your query result set that will be used for the parent cell in the Tree chart.
4. In the **Child field** drop-down list, select the name of the data column in your query result set that will be used as the child cell in the Tree chart.
5. Select the **Include a label** check box to include a label in each of the parent and child cells of the Tree chart.
6. In the **Label field** drop-down list, select the name of a data column in your query result set that will contain the label information for each parent and child cell in the tree chart.
You must select a data column from the drop-down list of available data columns that are in your query result set if you selected the **Include a label** check box.
7. In the **Number of levels to initially display** field, specify the maximum number of levels that are displayed initially in the tree chart under the parent objects. Type the number of levels in the field.
A value of 1 will display one level.
8. Click **Finish**.
The **Tree Chart Wizard** closes and the chart is displayed in the editor window.

Modifying the nodes state

You can modify the state of the Tree Chart or Organization Chart nodes on a visual dashboard.

Inserting an example chart

See the following example of adding an Organization Chart to a visual dashboard. This visual project is created on the basis of DEMO.EMPLOYEE table from the default Samples repository.

About this task

To create a visual project on the basis of DEMO.EMPLOYEE table:

Procedure

1. Create a visual dashboard.
2. Insert the query with the following SQL statement:


```
SELECT A.EMPLOYEE_ID, A.FULL_NAME, A.SUPERVISOR_ID
FROM DEMO.EMPLOYEE A
```
3. Add **Organization Chart** to the visual project.
4. On the second page of the **Organization Chart** wizard specify the following chart options:
 - a) Set SUPERVISOR_ID in the **Parent** field.
 - b) Set EMPLOYEE_ID in the **Child** field.
 - c) Set FULL_NAME in the **Label** field.
5. Click **Finish**. The example chart is added to the current visual project.

Saving the chart nodes state

You can save the chart nodes state in the Tree Chart or Organization Chart on a visual dashboard programmatically. See an example of saving the chart nodes state on the basis of the DEMO.EMPLOYEE example project.

Before you begin

You must have a visual dashboard with an added Organization Chart or Tree Chart.

About this task

To save the chart nodes state:

Procedure

1. In the DEMO.EMPLOYEE example project, add two global parameters **ExpandedNodesSaved** and **CollapsedNodesSaved** of the Text type.
2. Add two buttons to the project: **Button1** and **Button2**.
3. For **Button1** specify the **Caption** property as *Save*.
4. Add the action **Set Values** on the **Click** event. In the **Add New Action** wizard specify two actions:
 - a) `ExpandedNodesSaved=OrganizationChart1.DataLocator.ExpandedNodes`
 - b) `CollapsedNodesSaved=OrganizationChart1.DataLocator.CollapsedNodes`
5. For **Button2** specify the **Caption** property as *Load*.
6. Add the action **Set Values** on the **Click** event. In the **Add New Action** wizard specify two actions:
 - a) `OrganizationChart1.DataLocator.ExpandedNodes=ExpandedNodesSaved`
 - b) `OrganizationChart1.DataLocator.CollapsedNodes=CollapsedNodesSaved`

Note: At first, it is necessary to bind the value of **ExpandedNodes**, and then the value of the **CollapsedNodes** property.
7. Click **Runtime** tab. Expand and collapse the nodes. Click the **Save** button to write the nodes state to the global parameters.
8. Expand and collapse the nodes, click the **Load** button to restore the tree to the state saved by clicking the **Save** button.

Expanding or collapsing all chart nodes

You can expand or collapse all chart nodes in the tree or organization charts on a visual dashboard programmatically. See an example of expanding or collapsing the chart nodes on the basis of the DEMO.EMPLOYEE example project.

Before you begin

You must have a visual dashboard with an added Organization Chart or Tree Chart.

About this task

To expand or collapse all the chart nodes:

Procedure

1. In the DEMO.EMPLOYEE example project, add a button and set its **Caption** value to *Expand All*.
2. Add the action **Set Values** on the **Click** event with the expression:
 - a) `OrganizationChart1.DataLocator.ExpandedNodes=field("Query1", "EMPLOYEE_ID")`

Note: The separator in the expression depends on the locale settings.
3. Add a button and set its **Caption** value to *Collapse All*.
4. Add the action **Set Values** on the **Click** event with the expression:
 - a) `OrganizationChart1.DataLocator.CollapsedNodes=field("Query1", "EMPLOYEE_ID")`

Note: The separator in the expression depends on the locale settings.
5. Click **Runtime** tab and expand all chart nodes by clicking the **Expand All** and collapse them all by clicking **Collapse All**.

Note: If you add the example buttons to the Data Template they will work only with the node connected with this Data Template.

Expanding or collapsing all chart nodes by selecting their labels

You can expand or collapse the current chart node in the tree or organization chart on a visual dashboard by selecting their labels. See an example of expanding or collapsing the chart nodes on the basis of the DEMO.EMPLOYEE example project.

Before you begin

You must have a visual dashboard with an added Organization Chart or Tree Chart.

About this task

To expand or collapse a chart node:

Procedure

1. In the DEMO.EMPLOYEE example project, in the **Project Explorer** view expand the **OrganizationChart1** component, **DataTemplate1(Query1)**, **Level1** and select **Label1**.
2. Add the **Set Values** action for the **Click** event of the **Label1**:

```
OrganizationChart1.DataTemplate1.Level1.ChildLink1.Expanded=
!OrganizationChart1.DataTemplate1.Level1.ChildLink1.Expanded
```

3. Click **Runtime** tab and expand or collapse chart nodes by clicking their labels.

Expanding to a particular chart node

You can open a particular chart node in the tree or organization chart on a visual dashboard programmatically. See an example of expanding or collapsing the chart nodes on the basis of the DEMO.EMPLOYEE example project.

Before you begin

You must have a visual dashboard with an added Organization Chart or Tree Chart.

About this task

To open a particular chart node in the tree or organization chart:

Procedure

1. In the DEMO.EMPLOYEE example project, add the **Combo Box** object.
2. In the **Combo** wizard, specify the **Query1** that is used for the **OrganizationChart1**.
3. Add **EMPLOYEE_ID** column values to the **Combo Box**.
4. Expand **OrganizationChart1** component, **DataTemplate1(Query1)**, **Level1**, and then select **Label1**.
5. Modify its **Frame.FillStyle.FillColor** property by adding an expression:
=if(EMPLOYEE_ID=Combo1.Selection,#F00000,#78b1ea). To highlight the node after selecting it.

Note: The separator in the expression depends on the locale settings.

6. Add the **Set Values** action for the **Change** event of the **Combo Box** object:

```
OrganizationChart1.DataLocator.CollapsedNodes=field("Query1","EMPLOYEE_ID")
```

```
OrganizationChart1.DataLocator.ExpandedNodes=Combo1.Selection
```

```
OrganizationChart1.DataLocator.CollapsedNodes=Combo1.Selection
```

7. Click **Runtime** tab. Select the required IDs from the **Combo Box** object, the proper nodes will be expanded and highlighted with red.

Inserting a dial

Dials display points of data as a range between two numbers on a metered scale.

About this task

To insert a dial:

Procedure

1. Double-click the **Dial** object in the **Layouts** palette of the **Palette** view.
The **Dial Wizard** opens.
2. Select the minimum value that is displayed on the dial in the **Min** list.
3. Select the maximum value that is displayed on the dial in the **Max** list.
4. Select the starting angle of the dial in the **Start** list.
For example, specifying **-90** will display a horizontal line that stretches from the center of the dial to the left of the canvas.
5. Select the ending angle of the dial in the **End** list.
For example, specifying **90** will display a horizontal line that stretches from the center of the dial to the right of the canvas.
6. Select the **Symmetric** check box to automatically make the dial symmetrical.
This option sets the **End** angle as opposite to the **Start** angle. For example, if the **Start** angle is set to **-110**, the **End** angle is automatically set to **110**.
7. Select **Label Inside** to display the major tick mark labels in the inside of the arc of the dial.
8. Select **Label Outside** to display the major tick mark labels on the outside of the arc of the dial.
9. To add a sector to the dial, specify a value in the **Sectors** field and click the **Add New Entry** button.
The **Max** value is automatically changed to reflect the sum of the specified sectors. For example, as a default, there are three sectors set at **33.33** each. This creates a dial with three sectors, each a third of the total dial. If you add another sector with a value of **20**, the **Max** value is set to **120**.
10. To delete a sector, highlight an existing sector in the list and click the **Delete Entry** button.
11. Click **Finish**.
The **Dial Wizard** closes and the new dial object is displayed on the dashboard canvas.

Inserting a solid chart

Solid charts display your query results data in a visually rich format, like charts for all query types in QMF. In contrast to other composite charts that are available in visual projects, solid charts are added as the whole object to a scene. Solid charts are fully customizable.

About this task

To insert a solid chart object:

Procedure

1. Double-click the **Solid Chart** object in the **Layouts** palette of the **Palette** view. The **Solid Chart Wizard** opens.
2. Select a query for which you want to create a chart and click **Next**.
3. If the selected query has a saved chart, you can add it to your visual project on the **Chart** page, and click **Finish** to close the wizard.
4. To create a chart, on the **Chart type** page, specify the type of the chart. For more information about chart types, see [Specifying chart type options](#) topic. Click **Next**.
5. On the **Data series** page, specify the columns of the result set that you want to use to generate data points of a chart. For more information, see [Specifying data series options](#) topic. Click **Next**.

6. On the **Format** page, specify the options to define the visual output of the chart. For more information, see [Specifying format options](#) topic.
7. Click **Finish** to close the **Solid Chart Wizard**.
The created chart is displayed on the scene.

Editing a solid chart

You can edit the added solid chart to customize it for your particular purposes.

About this task

To edit a solid chart:

Procedure

1. In **Canvas** mode, right-click the solid chart that you want to edit and select **Edit Chart**.
2. In the "**Edit chart properties**" window, you can edit the following options:
 - Chart type
 - Data series
 - Chart format

For detailed information about editing the solid chart options see the topics:

- [Specifying chart type options](#)
 - [Specifying data series options](#)
 - [Specifying format options](#)
3. When you finish editing the solid chart options, click **OK** to apply your changes.

Changing a layout object's query

You can change a layout object's query by selecting **Change Query** in the **Project Explorer** menu.

About this task

To change a layout's object's query:

Procedure

1. In the **Project Explorer**, right-click the data template folder for the layout object whose query you want to change.
2. Select **Change Query** from the pop-up menu. The **Select Query** window opens where you can select a query that has been added to your **Queries** folder.
3. Click **OK**. The Layout object will use the new query that you selected.

Specifying a query parameter value

You can specify query parameter values for layout a object in the **Parameters** folder of the layout object's data template.

About this task

To specify a query parameter value:

Procedure

1. From the **Project Explorer** view, expand the data template folder for the layout object whose query parameter you want to set. Expand the **Parameters** folder.

2. Select the parameter from the **Parameters** folder. The name of the parameter and the current default value for the parameter are displayed in the **Properties** view.
3. Specify a new value for the parameter in the **Properties** view. You can specify a literal or an expression that will be evaluated at runtime.

Extending a layout object's functionality

Layout objects are the most versatile objects that you can insert in a visual project. As originally created by the layout wizard, the layout object is a simple, one dimensional representation of a single set of query results data. However, as you add functionality, a single layout object can become a dynamic, multidimensional representation of data obtained from more than a single query.

About extended layout objects

Extended layout objects allow you to roll up data across one or more columns in a result set.

Extended layout objects are different from standard layout objects in that they allow for a visual, client-side OLAP roll up of data across one or more columns in the associated query result set. This enables visual designers to create visual representations of data that are more organized and easy to use. Users can drill down into specific data symbols to obtain only the data they need, without having to view the entire result set at once. For example, the Q . STAFF table included in the sample repository has thirty-five rows, one for each employee in the company. A pie chart displaying the salary of all thirty-five employees would be visually cluttered and confusing. Rather than do this, the **Extended** option allows you to roll up salary data across one or more columns in the result set, providing on-demand drilldown to the underlying data. In the case of the Q . STAFF table, we can elect to roll the salary across department, followed by job type. Under this mode, the pie chart will display a slice for each department, with plus signs that let you expand each wedge to display each job type in that specific department. Expanding each job type slice then presents individual salary information for the specific department and specific job type. You can add as many drilldown operations to your layout objects as you want by specifying one or more columns to be moved to the side group in the query results or layout wizard.

Related concepts

Displaying OLAP data in layout objects

Displaying OLAP data in layout objects allows you to organize visual data into manageable hierarchies.

About data templates

Data templates visually represent query result set data in layout objects.

Related tasks

Changing the default data symbol

Each graphical layout object includes a default data symbol object which is used within the layout object to display each row of the query results data as a data point. For most layout objects, you can change the default data symbol and modify its properties.

Changing the default connector

Many graphical layout objects include a default connector object which is used within the layout to connect each data point. For those layout objects that include connectors, you can change the default connector and modify its properties.

Displaying multiple graphic objects for each data point

For most layout objects you can add additional graphics that will be displayed with each instance of the data symbol for each row of data that is included in the query result set.

Displaying multiple query results in a single layout

For many layout objects, you can display the results of multiple queries in a single layout object.

Copying an existing data template

You can add a data template to a layout object by copying an existing data template and then changing the query that is associated with the original data template.

Creating a new data template

You can use the **Insert Data template** command to add a new data template to a layout object.

Modifying a data template's visibility

You can specify whether a data template layer is displayed or not displayed in the layout object.

Nesting layout objects

When you nest layout objects, you can pass query result information from a higher level object to a lower level object. Because you have the ability to pass this information, you can use nested layouts to display more detailed information that relates to a specific data value.

Displaying OLAP data in layout objects

Displaying OLAP data in layout objects allows you to organize visual data into manageable hierarchies.

Certain layout objects allow you to display OLAP data. The layout objects that allow you to display OLAP data are:

- Bar Charts
- Column Charts
- Pie Charts

You can display OLAP data in these layouts by selecting an OLAP query from a multidimensional data source. The benefit of this is that it allows visual designers to choose along which dimensions they would like data to be displayed, and allows users to drill down into specific levels of detail. By progressively expanding portions of levels within a dimension, it is possible to drill-down into progressively more detail. At each level of detail, the layout object's data symbols refresh to visually represent that level's data points.

For example, you have a cube model that contains dimensions for product data, regional data, and time data. The product dimension contains individual products that are organized into four product families. The region dimension contains states that are organized into four geographic regions. The time dimension contains the four fiscal quarters of the current year. You want to create a column chart that displays profit totals for all products, and organize the visual data by time and region. You set the column chart to display data points from the PROFIT column as bars, and select the Time and Region dimensions to organize the data. The column chart is initially displayed as four bars that show the profit totals for each region for the whole year. Expandable cells that display the current dimensional levels are displayed under each column. The user can drill down into any region, and the column chart refreshes to display a column for every state in that region, while still displaying the other three regional totals. The user can then drill down further to display the quarterly totals for a specific state. Alternatively, the user can choose to drill down into a given region's year, displaying that region's quarterly totals. This dynamic organization allows users to quickly and easily make visual comparisons of data as it pertains to the larger data set.

Another way to organize OLAP data in a layout object is to use slicer objects. A slicer is used to filter the multidimensional data that is displayed in a layout object. A visual designer creates a slicer and selects a dimension of the layout object's OLAP query to filter. When the user clicks the slicer object in runtime, a window opens that lists all of the selected dimension's hierarchies and hierarchy elements. The user can then check which elements the layout object will display. Whenever the user changes the settings in the filter window, the layout object refreshes to reflect the change in filtered data.

For example, in the layout object mentioned above, the data is organized by time and region and always displays the product totals for each data symbol. You want to give users the option to select which particular product totals or product group totals are displayed at any given time. You insert a slicer object, select the Product dimension to filter the data, and connect the slicer object to the layout object using the **Connectivity** tool. When the user clicks the slicer object in runtime, a window opens that displays a tree containing all of the product dimension's hierarchy elements. When the user checks off the products and clicks **OK**, the layout object automatically refreshes to display only the totals for the selected products.

Related concepts

About extended layout objects

Extended layout objects allow you to roll up data across one or more columns in a result set.

About data templates

Data templates visually represent query result set data in layout objects.

Related tasks

Changing the default data symbol

Each graphical layout object includes a default data symbol object which is used within the layout object to display each row of the query results data as a data point. For most layout objects, you can change the default data symbol and modify its properties.

Changing the default connector

Many graphical layout objects include a default connector object which is used within the layout to connect each data point. For those layout objects that include connectors, you can change the default connector and modify its properties.

Displaying multiple graphic objects for each data point

For most layout objects you can add additional graphics that will be displayed with each instance of the data symbol for each row of data that is included in the query result set.

Displaying multiple query results in a single layout

For many layout objects, you can display the results of multiple queries in a single layout object.

Copying an existing data template

You can add a data template to a layout object by copying an existing data template and then changing the query that is associated with the original data template.

Creating a new data template

You can use the **Insert Data template** command to add a new data template to a layout object.

Modifying a data template's visibility

You can specify whether a data template layer is displayed or not displayed in the layout object.

Nesting layout objects

When you nest layout objects, you can pass query result information from a higher level object to a lower level object. Because you have the ability to pass this information, you can use nested layouts to display more detailed information that relates to a specific data value.

Changing the default data symbol

Each graphical layout object includes a default data symbol object which is used within the layout object to display each row of the query results data as a data point. For most layout objects, you can change the default data symbol and modify its properties.

About this task

To change a layout object's default data symbol:

Procedure

1. Double-click the layout to display the contents of its data template.
2. Select the existing data symbol and press DELETE.
3. Select the new data symbol that you want to use from the **Palette** view and insert it in the editor. The name of the new data symbol is added to the data template folder in the **Project Explorer** view.
4. You can use the **Properties** view to modify the properties of the data symbol, such as its shape or color.

Related concepts

About extended layout objects

Extended layout objects allow you to roll up data across one or more columns in a result set.

Displaying OLAP data in layout objects

Displaying OLAP data in layout objects allows you to organize visual data into manageable hierarchies.

About data templates

Data templates visually represent query result set data in layout objects.

Related tasks

Changing the default connector

Many graphical layout objects include a default connector object which is used within the layout to connect each data point. For those layout objects that include connectors, you can change the default connector and modify its properties.

Displaying multiple graphic objects for each data point

For most layout objects you can add additional graphics that will be displayed with each instance of the data symbol for each row of data that is included in the query result set.

Displaying multiple query results in a single layout

For many layout objects, you can display the results of multiple queries in a single layout object.

Copying an existing data template

You can add a data template to a layout object by copying an existing data template and then changing the query that is associated with the original data template.

Creating a new data template

You can use the **Insert Data template** command to add a new data template to a layout object.

Modifying a data template's visibility

You can specify whether a data template layer is displayed or not displayed in the layout object.

Nesting layout objects

When you nest layout objects, you can pass query result information from a higher level object to a lower level object. Because you have the ability to pass this information, you can use nested layouts to display more detailed information that relates to a specific data value.

Changing the default connector

Many graphical layout objects include a default connector object which is used within the layout to connect each data point. For those layout objects that include connectors, you can change the default connector and modify its properties.

About this task

To change a layout object's default connector:

Procedure

1. From the **Project Explorer**, select the default connector name from the data template folder and press **Delete**.
2. Select the new connector that you want to use from the **Palette** view and insert in the editor. The layout object that includes the new connector is presented in the editor window. The name of the new connector is added to the data template folder in the **Project Explorer**.
3. You can use the **Properties** view to modify the properties of the connector.

Related concepts

About extended layout objects

Extended layout objects allow you to roll up data across one or more columns in a result set.

Displaying OLAP data in layout objects

Displaying OLAP data in layout objects allows you to organize visual data into manageable hierarchies.

About data templates

Data templates visually represent query result set data in layout objects.

Related tasks

Changing the default data symbol

Each graphical layout object includes a default data symbol object which is used within the layout object to display each row of the query results data as a data point. For most layout objects, you can change the default data symbol and modify its properties.

Displaying multiple graphic objects for each data point

For most layout objects you can add additional graphics that will be displayed with each instance of the data symbol for each row of data that is included in the query result set.

Displaying multiple query results in a single layout

For many layout objects, you can display the results of multiple queries in a single layout object.

Copying an existing data template

You can add a data template to a layout object by copying an existing data template and then changing the query that is associated with the original data template.

Creating a new data template

You can use the **Insert Data template** command to add a new data template to a layout object.

Modifying a data template's visibility

You can specify whether a data template layer is displayed or not displayed in the layout object.

Nesting layout objects

When you nest layout objects, you can pass query result information from a higher level object to a lower level object. Because you have the ability to pass this information, you can use nested layouts to display more detailed information that relates to a specific data value.

Displaying multiple graphic objects for each data point

For most layout objects you can add additional graphics that will be displayed with each instance of the data symbol for each row of data that is included in the query result set.

About this task

For example, the default data symbol in a pie chart is a wedge. You can specify that each wedge of the pie chart additionally displays with a text label and two radio buttons. You can choose to add any of the graphic objects that are included in the **Palette** view with a few exceptions.

To add graphic objects for display with each row of query data:

Procedure

1. Double-click the layout object to display the contents of its data template.
2. Select the layout object that you want to add from the **Palette** view and insert it in the editor window. The new graphic object is drawn in the editor window. The name of the graphic object is added to the data template folder in the **Project Explorer**.
3. You can use the **Properties** view to modify the properties of the data symbol, such as its shape or color

Related concepts

About extended layout objects

Extended layout objects allow you to roll up data across one or more columns in a result set.

Displaying OLAP data in layout objects

Displaying OLAP data in layout objects allows you to organize visual data into manageable hierarchies.

About data templates

Data templates visually represent query result set data in layout objects.

Related tasks

Changing the default data symbol

Each graphical layout object includes a default data symbol object which is used within the layout object to display each row of the query results data as a data point. For most layout objects, you can change the default data symbol and modify its properties.

Changing the default connector

Many graphical layout objects include a default connector object which is used within the layout to connect each data point. For those layout objects that include connectors, you can change the default connector and modify its properties.

Displaying multiple query results in a single layout

For many layout objects, you can display the results of multiple queries in a single layout object.

Copying an existing data template

You can add a data template to a layout object by copying an existing data template and then changing the query that is associated with the original data template.

Creating a new data template

You can use the **Insert Data template** command to add a new data template to a layout object.

Modifying a data template's visibility

You can specify whether a data template layer is displayed or not displayed in the layout object.

Nesting layout objects

When you nest layout objects, you can pass query result information from a higher level object to a lower level object. Because you have the ability to pass this information, you can use nested layouts to display more detailed information that relates to a specific data value.

Displaying multiple query results in a single layout

For many layout objects, you can display the results of multiple queries in a single layout object.

About this task

For example, you could create a single XY chart that displays sales figures derived from one query and spending figures derived from another query. Each set of query results data that is drawn in a single layout object has its own data template. When you include multiple data templates (sets of query results) in a layout, each data template is drawn in the order in which it appears in the **Layout** folder in the **Project Explorer** view. The data template that appears last in the folder is drawn last and therefore appears on top of all previous layers.

You can add data templates to the following layout objects:

- **Candlestick Chart**
- **Event Band Chart**
- **Linear Map**
- **Scatter Chart**
- **Stock Chart**
- **Timeline Chart**
- **Column Chart**
- **Bar Chart**

Related concepts

About extended layout objects

Extended layout objects allow you to roll up data across one or more columns in a result set.

Displaying OLAP data in layout objects

Displaying OLAP data in layout objects allows you to organize visual data into manageable hierarchies.

About data templates

Data templates visually represent query result set data in layout objects.

Related tasks

Changing the default data symbol

Each graphical layout object includes a default data symbol object which is used within the layout object to display each row of the query results data as a data point. For most layout objects, you can change the default data symbol and modify its properties.

Changing the default connector

Many graphical layout objects include a default connector object which is used within the layout to connect each data point. For those layout objects that include connectors, you can change the default connector and modify its properties.

Displaying multiple graphic objects for each data point

For most layout objects you can add additional graphics that will be displayed with each instance of the data symbol for each row of data that is included in the query result set.

Copying an existing data template

You can add a data template to a layout object by copying an existing data template and then changing the query that is associated with the original data template.

Creating a new data template

You can use the **Insert Data template** command to add a new data template to a layout object.

Modifying a data template's visibility

You can specify whether a data template layer is displayed or not displayed in the layout object.

Nesting layout objects

When you nest layout objects, you can pass query result information from a higher level object to a lower level object. Because you have the ability to pass this information, you can use nested layouts to display more detailed information that relates to a specific data value.

Copying an existing data template

You can add a data template to a layout object by copying an existing data template and then changing the query that is associated with the original data template.

About this task

When you copy data templates, all child objects in the data template folder are also copied. As a result, you get an exact duplicate of the existing data template. You would then change the query that is associated with the new data template and you can optionally modify any child object properties.

To add a data template to a layout object by copying an existing data template:

Procedure

1. In the **Project Explorer** view, right-click the data template folder that you want to copy. Select **Copy** from the pop-up menu. The data template folder is copied to the clipboard.
2. In the **Project Explorer** view, right-click the layout object folder to which you want to add the data template. Select **Paste** from the pop-up menu. The data template folder is copied to the layout object's folder.
3. You can change the query that is associated with the data template in one of the following ways:
 - Right-click the new data template folder that you just added and select **Change Query**. The **Select Query** window opens where you can select a query that has been added to your **Queries** folder.
 - If the data template that you copied uses a query that has parameters, you can change the query for the data template by changing the value of the query parameter. Expand the data template folder that you just added and select the **Query Parameters** folder. Set the query parameter to a new value.
4. After a data template is added to the layout object, you can modify the objects and their properties as you require.
5. Click **Runtime** in the editor to view the results.

Related concepts

About extended layout objects

Extended layout objects allow you to roll up data across one or more columns in a result set.

Displaying OLAP data in layout objects

Displaying OLAP data in layout objects allows you to organize visual data into manageable hierarchies.

About data templates

Data templates visually represent query result set data in layout objects.

Related tasks

Changing the default data symbol

Each graphical layout object includes a default data symbol object which is used within the layout object to display each row of the query results data as a data point. For most layout objects, you can change the default data symbol and modify its properties.

Changing the default connector

Many graphical layout objects include a default connector object which is used within the layout to connect each data point. For those layout objects that include connectors, you can change the default connector and modify its properties.

Displaying multiple graphic objects for each data point

For most layout objects you can add additional graphics that will be displayed with each instance of the data symbol for each row of data that is included in the query result set.

Displaying multiple query results in a single layout

For many layout objects, you can display the results of multiple queries in a single layout object.

Creating a new data template

You can use the **Insert Data template** command to add a new data template to a layout object.

Modifying a data template's visibility

You can specify whether a data template layer is displayed or not displayed in the layout object.

Nesting layout objects

When you nest layout objects, you can pass query result information from a higher level object to a lower level object. Because you have the ability to pass this information, you can use nested layouts to display more detailed information that relates to a specific data value.

Creating a new data template

You can use the **Insert Data template** command to add a new data template to a layout object.

About this task

When you add a new data template to a layout object you have to add the graphic objects that will be used to display the data in that layer.

To add a new data template to a layout object:

Procedure

1. In the **Project Explorer** view, right-click the layout object folder to which you want to add the data template. Select **New > Data Template** from the pop-up menu. The **Data Template** wizard opens.
2. From the **Data Template** wizard select a query that will be used to obtain the data for this data template. The queries that are listed in the **Data Template** wizard are the ones that you have added to your **Queries** folder.
3. After a data template is added to the layout, you can modify the objects and their properties as you require. Select the data template folder that you added. A new canvas for the data template is opened in the editor window.
4. Click **Runtime** in the editor to view the results

Related concepts

About extended layout objects

Extended layout objects allow you to roll up data across one or more columns in a result set.

Displaying OLAP data in layout objects

Displaying OLAP data in layout objects allows you to organize visual data into manageable hierarchies.

About data templates

Data templates visually represent query result set data in layout objects.

Related tasks

Changing the default data symbol

Each graphical layout object includes a default data symbol object which is used within the layout object to display each row of the query results data as a data point. For most layout objects, you can change the default data symbol and modify its properties.

Changing the default connector

Many graphical layout objects include a default connector object which is used within the layout to connect each data point. For those layout objects that include connectors, you can change the default connector and modify its properties.

Displaying multiple graphic objects for each data point

For most layout objects you can add additional graphics that will be displayed with each instance of the data symbol for each row of data that is included in the query result set.

Displaying multiple query results in a single layout

For many layout objects, you can display the results of multiple queries in a single layout object.

Copying an existing data template

You can add a data template to a layout object by copying an existing data template and then changing the query that is associated with the original data template.

Modifying a data template's visibility

You can specify whether a data template layer is displayed or not displayed in the layout object.

Nesting layout objects

When you nest layout objects, you can pass query result information from a higher level object to a lower level object. Because you have the ability to pass this information, you can use nested layouts to display more detailed information that relates to a specific data value.

About data templates

Data templates visually represent query result set data in layout objects.

Data templates are used to create and control the visual representation of query result set data within layout objects. Anything that you place within a data template is replicated for each row of data that is returned from the query associated with the template. For example, the template for a line chart can consist of a data symbol and a connecting line. Each row of query data will then be represented by its own data symbol and connecting line. With direct access to the data template, content developers have the ability to construct a wide variety of customized charts and layouts. For example, one could produce a floating quartile graph by setting the start and end points of each bar in the data template and adding horizontal line primitives that are set to the mean and 25th and 75th percentiles of the data column(s).

Data templates provide tremendous control over a layout's behavior. However, you can also use them to make minor modifications to your charts and layouts. The following paragraphs provide a number of typical modifications that you can make to your data templates.

Customizing data symbols and graphic objects

You can customize the way query results are displayed in layout objects by editing the properties of the objects contained in their data templates. For example, you want to build an organization chart that displays the name of each member of a sales team in bold. By default, the text of the label objects that display each name is not bold. To display each name with bold text, you change the Bold property of the label object in the organization chart's data template folder to true. Now each member of the sales team's name will be displayed with bold text. You can also parameterize the display properties of data symbols and graphic objects. You want the names of sales team members who met their quotas to display as black text, and the names of sales team members who did not meet their quotas to display as red text. You change the Color property value of the label objects in the data template to a conditional expression. Now, when the organization chart is run, each name will display as either red or black depending on whether the sales team member met his or her quota.

Displaying multiple graphic objects for each data point

You can display multiple objects for each data point by adding multiple objects to the data template folder of the layout object. Using the example of an organization chart again, you might want to add a column chart that shows sales team member's quarterly sales totals by their name. You open the data template folder in the editor window and add a parameterized column chart object so that it is positioned directly beneath the label object of the organization chart. Now when the organization chart is run, a column chart is displayed for each team member that shows their individual quarterly sales totals. Now when the organization chart is run, a column chart is displayed for each team member that shows their individual quarterly sales totals.

Displaying multiple query result sets in a single layout object

A data template is always associated with a single query. However, you can draw from as many columns as you wish when presenting the data in your data template. For example, you wish to chart three query columns and set the color or size of each data point, based on three other query columns, and so on.

A layout can have multiple data templates, each of which draws data from differing queries, running against the same or differing data sources. When you add multiple data templates to a layout, such as a line chart, all data points are presented within the same pair of axes, auto-scaled (if set) to the minimum and maximum data values across all queries/data templates. Finally, you can mix the visual representation in a given layout. For example, you can present one or more line chart traces within a column chart, each running from the same or differing data templates.

Related concepts

About extended layout objects

Extended layout objects allow you to roll up data across one or more columns in a result set.

Displaying OLAP data in layout objects

Displaying OLAP data in layout objects allows you to organize visual data into manageable hierarchies.

Related tasks

Changing the default data symbol

Each graphical layout object includes a default data symbol object which is used within the layout object to display each row of the query results data as a data point. For most layout objects, you can change the default data symbol and modify its properties.

Changing the default connector

Many graphical layout objects include a default connector object which is used within the layout to connect each data point. For those layout objects that include connectors, you can change the default connector and modify its properties.

Displaying multiple graphic objects for each data point

For most layout objects you can add additional graphics that will be displayed with each instance of the data symbol for each row of data that is included in the query result set.

Displaying multiple query results in a single layout

For many layout objects, you can display the results of multiple queries in a single layout object.

Copying an existing data template

You can add a data template to a layout object by copying an existing data template and then changing the query that is associated with the original data template.

Creating a new data template

You can use the **Insert Data template** command to add a new data template to a layout object.

Modifying a data template's visibility

You can specify whether a data template layer is displayed or not displayed in the layout object.

Nesting layout objects

When you nest layout objects, you can pass query result information from a higher level object to a lower level object. Because you have the ability to pass this information, you can use nested layouts to display more detailed information that relates to a specific data value.

Modifying a data template's visibility

You can specify whether a data template layer is displayed or not displayed in the layout object.

About this task

The visibility property for a data template is a Boolean value that you can set to true or false. You can also enter a property expression that returns a true or false value.

To specify the visibility of a data template within a layout object:

Procedure

1. From the **Project Explorer** view, select the data template folder whose visibility property you would like to change. All of the properties for that data template are displayed in the **Properties** view.
2. Select **True** or **False** from the **Visibility** property list box or specify a property expression that returns **True** or **False**.

Results

Note: For layout objects that do not have a data template folder, visibility is a property of the layout object. You would select that layout object, and modify its **Visibility** property.

Related concepts

[About extended layout objects](#)

Extended layout objects allow you to roll up data across one or more columns in a result set.

[Displaying OLAP data in layout objects](#)

Displaying OLAP data in layout objects allows you to organize visual data into manageable hierarchies.

[About data templates](#)

Data templates visually represent query result set data in layout objects.

Related tasks

[Changing the default data symbol](#)

Each graphical layout object includes a default data symbol object which is used within the layout object to display each row of the query results data as a data point. For most layout objects, you can change the default data symbol and modify its properties.

[Changing the default connector](#)

Many graphical layout objects include a default connector object which is used within the layout to connect each data point. For those layout objects that include connectors, you can change the default connector and modify its properties.

[Displaying multiple graphic objects for each data point](#)

For most layout objects you can add additional graphics that will be displayed with each instance of the data symbol for each row of data that is included in the query result set.

[Displaying multiple query results in a single layout](#)

For many layout objects, you can display the results of multiple queries in a single layout object.

[Copying an existing data template](#)

You can add a data template to a layout object by copying an existing data template and then changing the query that is associated with the original data template.

[Creating a new data template](#)

You can use the **Insert Data template** command to add a new data template to a layout object.

[Nesting layout objects](#)

When you nest layout objects, you can pass query result information from a higher level object to a lower level object. Because you have the ability to pass this information, you can use nested layouts to display more detailed information that relates to a specific data value.

Nesting layout objects

When you nest layout objects, you can pass query result information from a higher level object to a lower level object. Because you have the ability to pass this information, you can use nested layouts to display more detailed information that relates to a specific data value.

About this task

For example, for each box in an employee Organization chart, you could also display a bar chart with the salary history for that particular employee.

To nest one layout object within another layout object:

Procedure

1. Double-click the layout to display the contents of its data template.
2. The canvas for that data template folder opens in the editor window.
3. Select the new layout object from the **Palette** view. Insert the new layout object in the editor window. The **[layout name] Wizard** opens. Specify the name of the query that will be used to obtain the data for this child layout. The queries that are listed in the **[layout name] Wizard** are the ones that you have added to your **Queries** folder for the visual project.
4. When you are creating the query that will be used by a child layout object you can do one of the following:
 - Create a query that references the parent layout object's query results using a query parameter. By referencing the parent query results, you can specify how to display your child layout.
 - Create a completely new query that does not reference the parent layout object's query results. If you do not reference the parent object query results, then your child object's complete query results set will display for every row of the parent object's query result set.
5. After the new layout object is inserted, you can modify its properties as you require.
6. Click **Runtime** in the editor to view the results.

Related concepts

About extended layout objects

Extended layout objects allow you to roll up data across one or more columns in a result set.

Displaying OLAP data in layout objects

Displaying OLAP data in layout objects allows you to organize visual data into manageable hierarchies.

About data templates

Data templates visually represent query result set data in layout objects.

Related tasks

Changing the default data symbol

Each graphical layout object includes a default data symbol object which is used within the layout object to display each row of the query results data as a data point. For most layout objects, you can change the default data symbol and modify its properties.

Changing the default connector

Many graphical layout objects include a default connector object which is used within the layout to connect each data point. For those layout objects that include connectors, you can change the default connector and modify its properties.

Displaying multiple graphic objects for each data point

For most layout objects you can add additional graphics that will be displayed with each instance of the data symbol for each row of data that is included in the query result set.

Displaying multiple query results in a single layout

For many layout objects, you can display the results of multiple queries in a single layout object.

Copying an existing data template

You can add a data template to a layout object by copying an existing data template and then changing the query that is associated with the original data template.

Creating a new data template

You can use the **Insert Data template** command to add a new data template to a layout object.

Modifying a data template's visibility

You can specify whether a data template layer is displayed or not displayed in the layout object.

Working with map objects

You can display Google Maps and add different objects to them in your visual dashboards.

About this task

The following Google Map objects are available:

- Google Map
- Google Map Marker
- Google Map Polyline
- Google Map Polygon
- Google Map Object

Inserting a Google Map object

You can use the **Google Map** object to display Google Maps with different objects on them in your visual dashboards.

About this task

To insert a Google Map object:

Procedure

1. In the **Palette** view, expand the **Map** palette and double-click the **Google Map** object to open the **Google Map Wizard**.
2. In the **Viewpoints** area, specify the start position of the **Google Map** object and its zoom level.
 - Select **Auto** check box if you want to set a map viewpoint automatically depending on the location of the objects that you added to the map.
 - Clear **Auto** check box and enter the coordinates in the **Latitude** and **Longitude** fields to specify the map center.
 - From the **Zoom level** list, select the zoom value for the **Google Map** object.
 - Select **Fixed** check box if you want to fix the map viewpoint. In this case, the users are not able to scroll a map or zoom it.

Note: When you select the **Fixed** check box, the Zoom control becomes unavailable in the **Google Map controls** area.

3. In the **Google Map controls** area, select the controls that you want to add to the **Google Map** object and click **Next**.

Note: The Google Map Wizard allows you to add three most frequently used controls to the Google Map object: Zoom, Map Type, and Scale. You can add other controls in the **Properties** view of your Google Map object. For more information, see the *Google Map* topic.

4. In the **Available map types** area, select the map types that are available for the users.
5. From the **Map type by default** list, select the map type that is loaded by default and click **Next**.
6. Define the content that you want to add to the **Google Map** object and click **Next**.

- Select **Create a map with a component** if you want to add an object to a map to display specific data, and then select the appropriate object from the **Select Component** area.
 - Select **Create an empty map** if you want to add a map without any objects on it.
7. Depending on the option that you selected, specify the required information on the next pages of the **Google Map Wizard** and click **Finish** to display the **Google Map** object that you created in the editor.

Note: To open the **Google Map** object that you created in QMF for WebSphere, you must obtain and specify the correct Google Map API Key.

Related tasks

Inserting a Google Map Marker object

You can use the **Google Map Marker** object to add markers to the **Google Map** object by specifying the coordinates of the markers.

Inserting a Google Map Marker object to display query results

You can use the **Google Map Marker** object to display query results as geographic markers on the **Google Map** object.

Inserting a Google Map Polyline object to display query results

You can use the **Google Map Polyline** object to display query results as a geographic map polyline.

Inserting a Google Map Polygon object to display query results

You can use the **Google Map Polygon** object in your visual dashboard to display query results as a geographic map polygon.

Inserting a Google Map Object to display query results

You can use the **Google Map Object** to display query results as a geographic map object.

Inserting a Google Map Marker object

You can use the **Google Map Marker** object to add markers to the **Google Map** object by specifying the coordinates of the markers.

Before you begin

The **Google Map Marker** object properties are defined in the *User's reference* section of the QMF help.

About this task

The **Google Map Markers** objects are interactive points on a map.

To add markers to the **Google Map** object:

Procedure

1. When you insert the **Google Map** object, select **Create a map with a component**, and then select the **Marker** option on the **Define map content** page of the **Google Map Wizard**. Then click **Next**.
2. Select **No** to enter the coordinates of the markers manually and click **Next**.
3. Define the markers on the **Define the properties for the markers** page.
 - In the **Caption** column, enter the name that you want to use for a marker.
 - In the **Tooltip** column, you can specify the text of a tooltip for a marker.
 - In the **Latitude** column, specify the latitude coordinate.
 - In the **Longitude** column, specify the longitude coordinate.
4. When you finish specifying the information for the markers, click **Finish** to close the **Google Map Wizard** and display the **Google Map** object with the markers that you added to it in the editor.

Tip: To add any other objects to the **Google Map** object that you created, drag them from the **Map** palette to the **Google Map** object in the **Project Explorer** view and specify the object properties.

Related tasks

[Inserting a Google Map object](#)

You can use the **Google Map** object to display Google Maps with different objects on them in your visual dashboards.

[Inserting a Google Map Marker object to display query results](#)

You can use the **Google Map Marker** object to display query results as geographic markers on the **Google Map** object.

[Inserting a Google Map Polyline object to display query results](#)

You can use the **Google Map Polyline** object to display query results as a geographic map polyline.

[Inserting a Google Map Polygon object to display query results](#)

You can use the **Google Map Polygon** object in your visual dashboard to display query results as a geographic map polygon.

[Inserting a Google Map Object to display query results](#)

You can use the **Google Map Object** to display query results as a geographic map object.

Inserting a Google Map Marker object to display query results

You can use the **Google Map Marker** object to display query results as geographic markers on the **Google Map** object.

Before you begin

The **Google Map Marker** object properties are defined in the *User's reference* section of the QMF help.

About this task

The **Google Map Markers** objects are interactive points on a map.

To display query results as markers:

Procedure

1. When you insert the **Google Map** object, select **Create a map with a component**, and then select the **Marker** option on the **Define map content** page of the **Google Map Wizard**. Then click **Next**.
2. Select **Yes** to display query results as geographic markers on the **Google Map** object and click **Next**.
3. In the **All Data Sources** tree, select a table or query, which results you want to use to define markers on the map, and click **Next**.
4. From the **Latitude field** list, select a column to supply the latitude coordinate for the map markers.
5. From the **Longitude field** list, select a column to supply the longitude coordinate for the map markers.
6. From the **Caption field** list, select a column to populate the labels for the map markers.
7. Select the **Enable information window** check box to allow the users to open an information window when they click the markers on the map.
8. From the **Information field** list, select a column to populate the information windows of the map markers.
9. Click **Finish** to close the **Google Map Wizard** and display the **Google Map** object with the markers that you added to it in the editor.

Tip: To add any other objects to the **Google Map** object that you created, add a new **Data Template** to it, and then drag the required object from the **Map** pallet to the **Data Template** in the **Project Explorer** view.

Related tasks

[Inserting a Google Map object](#)

You can use the **Google Map** object to display Google Maps with different objects on them in your visual dashboards.

[Inserting a Google Map Marker object](#)

You can use the **Google Map Marker** object to add markers to the **Google Map** object by specifying the coordinates of the markers.

Inserting a Google Map Polyline object to display query results

You can use the **Google Map Polyline** object to display query results as a geographic map polyline.

Inserting a Google Map Polygon object to display query results

You can use the **Google Map Polygon** object in your visual dashboard to display query results as a geographic map polygon.

Inserting a Google Map Object to display query results

You can use the **Google Map Object** to display query results as a geographic map object.

Inserting a Google Map Polyline object to display query results

You can use the **Google Map Polyline** object to display query results as a geographic map polyline.

Before you begin

The **Google Map Polyline** object properties are defined in the *User's reference* section of the QMF help.

About this task

The **Google Map Polyline** object creates a linear overlay on Google Maps. It consists of a series of points and creates a series of line segments that connect the points in the ordered sequence.

To insert a Google Map Polyline object:

Procedure

1. When you insert the **Google Map** object, select **Create a map with a component**, and then select the **Polyline** option on the **Define map content** page of the **Google Map Wizard**. Then click **Next**.
2. In the **All Data Sources** tree, select a table or query, which results you want to use to define a polyline on the map, and click **Next**.
3. Select the **Direct** check box to define the latitude and longitude coordinates of the polyline points and click **Next**.
 - a) From the **Tooltip field** list, select a column to supply a tooltip text for the polyline on the map.
 - b) From the **Vertices field** list, select a column to supply the coordinates for the polyline points.
 - c) From the **Point format** list, select the format for coordinates (Latitude-Longitude or Longitude-Latitude).
4. Select the **Encoded** check box to define a polyline in the Google Maps encoded format and click **Next**.
 - a) From the **Tooltip field** list, select a column to supply a tooltip text for the polyline on the map.
 - b) Select the columns to specify a series of points within a polyline from the **Points field** list.

Tip: To increase the performance of your visual dashboard, you can use the encoded polyline algorithm format for your polyline and polygon objects. You can obtain more information about it from the Google Developers website.

5. Click **Finish** to close the **Google Map Wizard** and display the **Google Map** object with the polyline that you added to it in the editor.

Tip: To add any other objects to the **Google Map** object that you created, add a new **Data Template** to it, and then drag the required object from the **Map** pallet to the **Data Template** in the **Project Explorer** view.

Related tasks

Inserting a Google Map object

You can use the **Google Map** object to display Google Maps with different objects on them in your visual dashboards.

Inserting a Google Map Marker object

You can use the **Google Map Marker** object to add markers to the **Google Map** object by specifying the coordinates of the markers.

Inserting a Google Map Marker object to display query results

You can use the **Google Map Marker** object to display query results as geographic markers on the **Google Map** object.

Inserting a Google Map Polygon object to display query results

You can use the **Google Map Polygon** object in your visual dashboard to display query results as a geographic map polygon.

Inserting a Google Map Object to display query results

You can use the **Google Map Object** to display query results as a geographic map object.

Inserting a Google Map Polygon object to display query results

You can use the **Google Map Polygon** object in your visual dashboard to display query results as a geographic map polygon.

Before you begin

The **Google Map Polygon** object properties are defined in the *User's reference* section of the QMF help.

About this task

Like the **Google Map Polyline** object, the **Google Map Polygon** object represents a series of sequential data points. However, instead of being open-ended, polygons are designed to define regions within a closed loop.

You can define the characteristics of the polygon edge and the fill area within the enclosed region.

To insert a Google Map Polygon object:

Procedure

1. When you insert the **Google Map** object, select **Create a map with a component**, and then select the **Polygon** option on the **Define map content** page of the **Google Map Wizard**. Then click **Next**.
2. In the **All Data Sources** tree, select a table or query, which results you want to use to define a polygon on the map, and click **Next**.
3. Select the **Direct** check box to define the latitude and longitude coordinates of the polygon points and click **Next**.
 - a) From the **Tooltip field** list, select a column to supply a tooltip text for the polygon on the map.
 - b) From the **Vertices field** list, select a column to supply the coordinates for the polygon points.
 - c) From the **Point format** list, select the format for coordinates (Latitude-Longitude or Longitude-Latitude).
4. Select the **Encoded** check box to define a polygon in Google Maps encoded format and click **Next**.
 - a) From the **Tooltip field** list, select a column to supply a tooltip text for the polygon on the map.
 - b) Select the columns to specify a series of points for a polygon from the **Points field** list.

Tip: To increase the performance of your visual dashboard, you can use the encoded polyline algorithm format for your polyline and polygon objects. You can obtain more information about it from the Google Developers website.

5. Click **Finish** to close the **Google Map Wizard** and display the **Google Map** object with the polygon that you added to it in the editor.

Tip: To add any other objects to the **Google Map** object that you created, add a new **Data Template** to it, and then drag the required object from the **Map** pallet to the **Data Template** in the **Project Explorer** view.

Related tasks

[Inserting a Google Map object](#)

You can use the **Google Map** object to display Google Maps with different objects on them in your visual dashboards.

[Inserting a Google Map Marker object](#)

You can use the **Google Map Marker** object to add markers to the **Google Map** object by specifying the coordinates of the markers.

[Inserting a Google Map Marker object to display query results](#)

You can use the **Google Map Marker** object to display query results as geographic markers on the **Google Map** object.

[Inserting a Google Map Polyline object to display query results](#)

You can use the **Google Map Polyline** object to display query results as a geographic map polyline.

[Inserting a Google Map Object to display query results](#)

You can use the **Google Map Object** to display query results as a geographic map object.

Inserting a Google Map Object to display query results

You can use the **Google Map Object** to display query results as a geographic map object.

Before you begin

The **Google Map Object** properties are defined in the *User's reference* section of the QMF help.

About this task

To display query results as a geographic map object:

Procedure

1. When you insert the **Google Map** object, select **Create a map with a component**, and then select the **Object** option on the **Define map content** page of the **Google Map Wizard**. Then click **Next**.
2. In the **All Data Sources** tree, select a table or query which results you want to use to define the object on the map.
3. From the **Tooltip field** list, select a column to supply the text of a tooltip for the object.
4. From the **Bounds field** list, select a column to supply the boundaries of the object on the map.
5. From the **Content field** list, select a column with the content that you want to display on the map, for example an image.
6. Click **Finish** to close the **Google Map Wizard** and display the **Google Map** object with the object that you added to it in the editor.

Tip: To add any other objects to the **Google Map** object that you created, add a new **Data Template** to it, and then drag the required object from the **Map** pallet to the **Data Template** in the **Project Explorer** view.

Related tasks

[Inserting a Google Map object](#)

You can use the **Google Map** object to display Google Maps with different objects on them in your visual dashboards.

[Inserting a Google Map Marker object](#)

You can use the **Google Map Marker** object to add markers to the **Google Map** object by specifying the coordinates of the markers.

[Inserting a Google Map Marker object to display query results](#)

You can use the **Google Map Marker** object to display query results as geographic markers on the **Google Map** object.

[Inserting a Google Map Polyline object to display query results](#)

You can use the **Google Map Polyline** object to display query results as a geographic map polyline.

[Inserting a Google Map Polygon object to display query results](#)

You can use the **Google Map Polygon** object in your visual dashboard to display query results as a geographic map polygon.

Working with the Layers view

The **Layers** view allows you to group palette objects into separate layers that can be displayed or hidden during design time.

About this task

Layers aid the design process by allowing you to display and hide selected objects that overlap each other on a scene's canvas. Layers are grouped together within each level of detail of a scene, and each palette object is initially placed in the **Default** layer. Any number of layers can be added to each level of detail. To open the **Layers** view:

Procedure

1. Open a visual dashboard and select a level of detail of a scene.
2. Select **Window > Show View > Layers**.
The **Layers** view opens and the current level of detail's layers are displayed.
3. To add a layer to the current level of detail, right-click the level in the **Layers** view and select **Insert Layer**.
A new layer is added to the tree.
4. To rename a layer, right-click the layer, select **Rename Layer [Layer Name]**, and specify a new name for the layer.
5. To delete a layer, right-click the layer and select **Delete**.
The layer is deleted.
6. To add a palette object to an existing layer, right-click the palette object on the scene canvas, select **Layer**, and select the layer that you need from the pop-up menu.
The palette object is added to the specified layer in the **Layers** view.
7. To move a palette object to a different layer, in the **Layers** view, click and drag the object to the layer.
8. To hide a layer's contents, clear individual objects or the layer itself in the **Layers** view.
If a layer is cleared, all of the objects that are assigned to the layer are hidden on the scene canvas. If an individual object is cleared, only that object will be hidden.

Working with global resources

Globals are resources that can be used by all the pages of a visual report or all the scenes of a visual dashboard.

About this task

You can use globals for computing object properties. All global resources are displayed in the **Globals** folder of the **Project Explorer**. You can choose to have one or more of the following globals available to the entire visual project:

Parameters

Parameters are available to your entire visual project and enable you to use variable values when specifying object properties.

Color Maps

A color map can be used to vary the color of a visual project object depending on a value associated with the object.

Color Sequences

A color sequence can be used to modify the color properties of a visual project object based on an index value.

Security lists

Security lists can be used to tailor visual report or dashboard content to the security level of the user who is viewing the report or dashboard.

Images

A stock image enables you to store your image resources with your visual project. Storing image resources with your visual project increases the portability and performance of the project.

Expiration Schedules

Expiration schedules can be set up to dictate when given data expires.

JavaScript Modules

JavaScript Modules enable you to use custom JavaScript functions in visual projects.

Translation Tables

Translation tables enable dashboard designers to publish a single visual dashboard that can be viewed in multiple languages.

Working with color sequences

A color sequence is a type of global resource that can be used to modify the color properties of a visual project object based on an index value. Each index value specifies which wedge in a color sequence wheel will contain the color for an object.

About this task

You are given a default color sequence that includes set colors in the color sequence wheel. You can also set up unique color sequences that will vary the colors that are included in the color sequence wheel. You specify that a visual project object will use a color sequence when you are specifying the properties of that object. At that time you also decide whether you will specify the index value for a color sequence with an explicit number, an expression or a function.

Creating a new color sequence

You use the **New Color Sequence** wizard to create a color sequence for your visual project.

About this task

To create a new color sequence:

Procedure

1. You open the **New Color Sequence** wizard in one of the following ways:
 - From the **Project Explorer** view, click the **Globals** node. Right-click the **Color Sequences** node. Select **Insert Color Sequence** from the pop-up menu. The **New Color Sequence** wizard opens.
 - From the **Project Explorer** view, click the **Globals** node. Double-click the **Color Sequences** node. The **New Color Sequence** wizard opens.
2. Type a unique name for the color sequence in the **Name** field.
3. The colors that will appear in the color sequence are displayed in the **Colors** wheel. Clicking on each wedge will display the wedge position number and its associated index value. For example, the first wedge displays a zero (0) and will represent the color that is displayed for an index value of zero (0).
4. To change a wedge color, click the wedge, then select a new color by clicking the color button. The **Color** window opens where you can select a new color. Click **OK**. The selected wedge is displayed with the new color. You repeat this process for each wedge color that you want to change.
5. To undo any changes that you have made, click the **Undo changes** button. The color of the selected wedge returns to its previous color.

6. To move a selected wedge forward in the wheel, click the wedge and then click the **Move Forward** button. The wedge moves forward one slot in the wheel.
7. To move a selected wedge backward in the wheel, click the wedge and then click the **Move Back** button. The wedge moves backward one slot in the wheel.
8. To add additional wedges to the pie, click the **Add** button. A new wedge is added to the pie in the last position. It will be added using the color of the currently selected wedge.
9. To remove a wedge, select the wedge that you want to remove from the pie, and click the **Remove** button.
10. Click **Finish**. The **New Color Sequence** wizard closes and the color sequence is displayed in the **Color Sequences** folder.

Using a color sequence

To use a color sequence specify the name of a color sequence in a color property cell for an object that you insert in your visual project.

About this task

To use a color sequence:

Procedure

1. Insert an object in your visual project.
2. From the most convenient view, (Properties, Outline or the Project Explorer), click the color property that will use the color sequence.
3. Specify the name of the color sequence that you want to use and the index value. There are several ways that you can specify the index value:

- Literally: You can use a specific number to indicate the index value. For example:

```
colorSeq(Colorseq1, 4)
```

The color of this object will be the same color as the wedge labeled index 4 in the color wheel of the color sequence named Colorseq1.

- With an expression: You can create an expression that can be used to calculate the index value. For example:

```
colorSeq(Colorseq1, value1 + 5)
```

The color of this object will be from the color sequence named Colorseq1. The index value will be calculated by adding 5 to the value of value1.

- Using a function: You can use a function to generate an index value. For example:

```
colorSeq(Colorseq1, PointNumber())
```

The PointNumber() function will be used with a color sequence named Colorseq1. The PointNumber() function assigns a unique color to the object for each row returned by a query.

Working with color maps

A color map is a type of global resource that can be used to vary the color of a visual project object depending on a value associated with the object.

About this task

For example, you might want a bar chart to display bar colors depending on the values returned by the query. If you want the color to change from red to black when your product shows a profit, you can create a color map that will specify the colors to use when the data reaches certain breakpoint values. Color

maps are either discrete or blended. Discrete color maps display a different color for each data value. Blended color maps blend the colors. Breakpoints specify the value at which the color changes.

Creating a color map

You use the **New Color Map** wizard to create a color map for your visual project.

About this task

To create a color map:

Procedure

1. You open the **New Color Map** wizard in one of the following ways:

- From the **Project Explorer** view, click the **Globals** node. Right-click the **Color Maps** node. Select **Insert Color Map** from the pop-up menu. The **New Color Map** wizard opens.
- From the **Project Explorer** view, click the **Globals** node. Double-click the **Color Maps** node. The **New Color Map** wizard opens.

2. Type a unique name for the color map in the **Name** field.

3. Click **Distribute**.

The **Color Map Breakpoints** window opens.

4. Specify the total number of breakpoints in the **Breakpoints** list.

5. Specify the minimum breakpoint data value in the **Min** value field.

6. Type the maximum breakpoint data value in the **Max** value field.

7. Select the colors you want from the **Colors** list box.

You can choose to set the colors to black, to the colors of the rainbow, or to shades of gray.

8. Click **OK**.

The **Color Map Breakpoints** window closes and the number of breakpoints that you specified are displayed.

9. Specify the properties of each breakpoint.

a) Select a breakpoint from the list.

The breakpoint value and color are displayed.

b) Specify the breakpoint value in the **Breakpoint** field.

c) Click the **Active color** box.

The **Color** window opens.

d) Select the color that you want to display when the breakpoint is reached and click **OK**.

The **Color** window closes and the new color is displayed in the **Active color** box.

e) Click **Set**.

The new value and color are set.

f) Repeat this process for each breakpoint in the list.

10. For all color maps, click **First color** to select the first color of the color map. The first color is the color used for all data values that are less than your first breakpoint value.

11. Specify the type of color transition that you want using the **Color Transitions** radio buttons. Your choices are:

- Select **Discrete** to display a pure color from one breakpoint data value to the next breakpoint data value.
- Select **Blend** to blend the colors from one breakpoint data value color to the next breakpoint data value color.

12. If you have selected to blend the colors from one breakpoint data value to the next breakpoint data value, then you can select a last color for your color map. The last color is the color used for all data

values that are greater than your last breakpoint. The last color and the first color appear un-blended in the color map. To select the last color of a blended color map, click **Last color**.

13. Click **Finish**.

The **New Color Map** wizard closes and the color map is displayed in the **Color Maps** folder.

Using a color map

To use a color map specify the name of a color map in a color property cell for an object that you insert in your visual project.

About this task

To use a color map:

Procedure

1. Insert an object in your visual project.
2. From the most convenient view (Properties, Outline or the Project Explorer), click the color property that will use the color map.
3. Specify the name of the color map that you want to use and a data value. There are several ways that you can specify the data value:

- Literally: You can use a specific number to indicate the data value. For example:

```
colorMap(Colormap1, 10)
```

The data value 10 will be compared against the breakpoints that have been specified for the color map named Colormap1. The color of this object will be determined by the color assigned to a data value of 10.

- With an expression: You can create an expression that can be used to calculate the data value. For example:

```
colorMap(Colormap1,(close_price - open_price))
```

The expression will be evaluated based on the current data values. The calculated data value will be compared against the breakpoints that have been specified for the color map named Colormap1. The color of this object will be determined by the color assigned to the calculated data value.

- Using a function: You can use a function to generate a data value. For example:

```
colorMap(Colormap1,PointNumber())
```

The PointNumber() function will be used with a color map named Colormap1. The PointNumber() function assigns a unique color to the object for each row returned by a query.

Creating a security list

Security lists can be used to tailor visual report or dashboard content to the security level of the user who is viewing the report or dashboard.

About this task

For example, members of the sales team could see sales data, whereas members of the executive team could see sales and profitability data on a particular chart in the dashboard. You can only implement the security list function for visual reports and dashboards that are saved in secured repositories.

To create a security list and implement the security list function:

Procedure

1. You can open the **New Security List Wizard** in one of the following ways:

- From the **Project Explorer** view, expand the **Globals** node. Right-click the **Security Lists** node. Select **Insert Security List** from the pop-up menu. The **New Security List Wizard** opens.
 - From the **Project Explorer** view, expand the **Globals** node. Double-click the **Security Lists** node. The **New Security List Wizard** opens.
2. Type a unique name for the security list in the **Name** field.
 3. Select the users and groups that will be a part of this security list.
 4. Click **Finish**. The security list is added to the **Security Lists** folder in the **Project Explorer** view. You can create multiple security lists. Each list can contain a similar or unique set of users and groups. Each list must have a unique name.
 5. Insert an object into your visual report or dashboard.
 6. Click in the cell of the property that you will control based on the security level of your user. You can choose any property. In order to control what data is displayed to a user, you would select the visibility property for the object.
 7. Click the **Edit with Expression Designer** toolbar button in the **Properties** view. The **Expression Designer** opens. Clear the designer work area. Expand the **Security** node of the function tree. Double-click the function **isSecurityLevel**. The template for the function appears as follows in the designer work area: `isSecurityLevel(strText)`.
 8. You will replace the text parameter with a security list. Highlight `strText`. Expand the **Security List** folder from the designer tree. Double-click the security list that you want to view the object. The property value appears in the designer work area as follows: `isSecurityLevel("SecurityList1")`.
 9. Click **Apply**.
 10. Click **OK**. Only the users that are included in the security list that you selected can view the object.

Storing images with visual projects

A stock image is a type of global resource that enables you to store your image resources with your visual project.

About this task

Storing images with your visual project increases the portability and performance of the project. Portability of the project is improved because you can run the project on any machine and still have access to your image as it comes along with the project. Performance improves because images do not have to be loaded or reloaded each time they are used.

For example, the processing overhead of loading and reloading a corporate logo that is used on every page of a visual report or in every scene of a visual dashboard could be significant. Using a stock image for the corporate logo cuts the processing time of loading and reloading the image. Pages of a visual report and scenes from a visual dashboard are displayed to your users more quickly. In addition, a stock image of the corporate logo eliminates the need for you to hard code the image location in your visual project.

The following image file formats can be stored as stock images:

- Bitmap (.bmp)
- Device-independent bitmap (.dib)
- Graphic Interchange format (.gif)
- Joint Photographic Experts Group (.jpeg)
- Portable Network Graphics (.png)

Creating a stock image

You can create a stock image for each image that you want to use in your visual project.

About this task

To create a new stock image:

Procedure

1. You can open the **Open** window in one of the following ways:
 - From the **Project Explorer** view, click the **Globals** node. Right-click the **Images** node. Select **Insert Image** from the pop-up menu. An operating system specific **Open** window opens.
 - From the **Project Explorer** view, click the **Globals** node. Double-click the **Images** node. An operating system specific **Open** window opens.
2. From the **Open** window search for and select the image that you want to use in your visual project.
3. Click **Open**. A stock image global resource is created. The stock image name is the same as the image name. The new stock image is listed in the **Images** node in the **Project Explorer** view.

Using a stock image

You must associate the stock image in the **Images** folder with the **Picture** object in your visual project.

About this task

You can associate a stock image to the **Picture** object in one of the following ways:

- From the **Project Explorer** view, click a stock image that you have added to the **Images** folder and drag the stock image to your visual project canvas. A **Picture** object is created and the **ImageSource** property is displayed in the **Properties** view and is automatically updated with the name of the stock image.
- From the **Palette** view, select the **Picture** object and insert it in your visual project. In the **Properties** view, click the **ImageSource** property cell. Type the name of the stock image using the **stockImage()** function as follows: `=stockImage("My_Image")` where `My_Image` is the name of a stock image that you have added to the **Images** folder for the project.

The picture in your visual project displays the specified stock image. You can then modify additional object properties such as image scaling.

Specifying variable values using parameters

Parameters are a way of specifying variable values for use by your visual project.

About this task

Parameters enable you to do the following tasks:

- Capture user input.
- Capture current context, such as query results, object property values, or locations.
- Pass user input or context information from one part of a dashboard to another.
- Vary query results based on dynamic criteria.
- Vary navigation results based on dynamic criteria.
- Vary visual project design elements based on dynamic criteria.

The following types of parameters are available:

Global Parameters

A parameter that is available to your entire visual project and to the user of your visual dashboard. You can pass global parameters when you specify object properties or when you create event actions. You also have the option of making global parameters public at runtime so they are accessible to the user.

Scene Parameters

A parameter that is available only to a specific scene in a visual dashboard. You can pass scene parameters when you create embedded scenes and event actions.

Query Parameters

Query parameters contain the value that will be sent to the query and used at runtime.

You can use parameters in many ways. Event actions typically capture user input in a parameter. You can use global and scene parameters when you design interactive controls in your scenes; when a user makes a selection in the control, an underlying event action sets the parameter value. The parameter value determines what data is displayed or how it is displayed.

Global, scene, and query parameters contain the following attributes:

- Name (parameter names must be unique)
- Data type
- Description (optional)
- Default value (optional)

Note: You should provide a default value for your parameters so you can test them.

Working with global parameters

A global parameter is a type of global resource that is available to your entire visual project. Global parameters enable you to use variable values within your visual dashboard.

Global parameters can either be public or private. A private global parameter is set exclusively within the visual dashboard. A public parameter can be modified by the user at runtime.

QMF for Workstation provides the following global parameters:

- **DeviceIdentifier** is a global parameter, which stores the UUID for QMF for WebSphere or MAC address for QMF for Workstation. When the LoadComplete event occurs, the value of this parameter is defined automatically .

Note: In QMF for WebSphere, the UUID is saved to cookies, but if cookies are disabled, the application gets a new UUID every time when the LoadComplete event occurs.

- **ExternalEventID** is a global parameter, set to a particular control on a scene. It saves **EventID** from **Send to Embedded Scene** event action.
- **Geolocation** is a global parameter that stores the device geolocation data. To get the data, run the [Refresh geolocation](#) event action.
- **Translation Table** defines the translation table that is used to display the text values in the visual dashboard.
- **ViewerHeight** contains the height of the current visible part of the scene. This parameter is read-only and its value depends on the default measurement units. This global parameter has been defined for developing visual dashboards. You can use it in expressions when it is convenient to take as a parameter the current height of the visible part of the scene. This parameter value and the expressions with it are automatically updated as you change the size of the current visible scene.
- **ViewerWidth** contains the width of the current visible part of the scene. This parameter is read-only and its value depends on the default measurement units. This global parameter has been defined for developing visual dashboards. You can use it in expressions when it is convenient to take as a parameter the current width of the visible part of the scene. This parameter value and the expressions with it are automatically updated as you change the size of the current visible scene.
- **ViewerX** defines the viewer's current x-coordinate location, in inches. The default value is zero (0) which translates to the center of the screen. This parameter value is automatically updated as a user repositions in the visual project. This global has been defined for use by visual dashboards.
- **ViewerY** defines the viewer's current y-coordinate location, in inches. The default value is zero (0) which translates to the center of the screen. This parameter value is automatically updated as a user repositions in the visual project. This global has been defined for use by visual dashboards.
- **ViewerZoom** defines the viewer's current zoom level. The default value is 100%. This parameter value is automatically updated as the user changes zoom levels.

You can define your own global parameters.

Adding a global parameter

You can define your own global parameters.

About this task

To add a user-defined global parameter:

Procedure

1. You can open the **Insert Parameter** window in one of the following ways:
 - From the **Project Explorer** view, expand the **Globals** node. Right-click the **Parameters** node. Select **Insert Parameter** from the pop-up menu.
 - From the **Project Explorer** view, expand the **Globals** node. Double-click the **Parameters** node. The **Insert Parameter** window opens.
2. Type a unique name for the global parameter in the **Name** field.
3. Select a data type from the **Data type** list box. The data type will describe what type of data will be included in the parameter. Your choices are:
 - **Boolean**: True or False
 - **Color**: Color
 - **DateTime**: Date and time value
 - **DateTimeSet**: Multiple data and time values
 - **FilePath**: Path to a file
 - **FontName**: Valid font type
 - **Integer**: Whole number
 - **Literal**: As is data. This data type only applies to query parameters.
 - **Number**: Floating point number
 - **NumberSet**: Multiple floating point numbers
 - **Percentage**: Value that will be used as a percentage
 - **Point**: Single x,y point value
 - **PointSet**: Multiple x,y point values
 - **PolySet**: Multiple point variables that can be used to construct a polygon
 - **Text**: A text string
 - **TextSet**: Multiple text strings
4. To add a description, type the text in the **Description** field.
5. To specify that the parameter be available to the user, select the **Public at runtime** check box. When checked, a **Runtime Settings** window will be issued at runtime and the user can enter information.
6. To specify a default value, select the **Has default value** check box. A field is displayed where you can type a default value for the global parameter.

Using global parameters

Global parameters can be used to specify object properties in visual projects.

Before you begin

About this task

To use global parameters when specifying object properties:

Procedure

1. Insert an object in your visual project.

2. From the **Properties** view, select the property that will derive its value using your global parameter and type =your_global_parameter_name in the corresponding property cell.
3. You can also use a global parameter in an expression that will be evaluated in order to derive the property value. Open the **Expression Designer**. The global parameters that have been defined are listed in the left pane.

Working with Expiration Schedules

Use expiration schedules to specify when the data that is related to your visual project must expire.

About this task

Expiration schedules can be set up to automatically refresh the data that is called in a visual project. If an expiration schedule is assigned to a query in a visual project, the query results that are obtained from the query are saved for the given amount of time that the schedule dictates. When a user runs the visual report or dashboard, the query results that are saved in the cache are used and the query is not rerun. Once the time and date set up by the schedule is reached, the query results expire, and the next time a user runs the visual report or dashboard, the query is rerun. Expiration schedules can save time and database resources by only running queries when updated data is needed.

Note: You can use expiration schedules only for the visual projects that are saved to repository.

To create a new expiration schedule:

Procedure

1. The **Expiration Schedule Configuration** wizard can be opened in one of the following ways:
 - From the **Project Explorer** view, expand the **Globals** node. Right-click the **Expiration Schedules** node. Select **Insert Schedule** from the pop-up menu. The **Expiration Schedule Configuration** wizard opens.
 - From the **Project Explorer** view, expand the **Globals** node. Double-click the **Expiration Schedules** node. The **Expiration Schedule Configuration** wizard opens.
2. Enter a name for the expiration schedule in the **Schedule name** field
3. Select how often the schedule expires in the **Data expires** radio group. Valid choices are:
 - **hourly** to have the schedule expire every hour.
 - **daily** to have the schedule expire every day.
 - **weekly** to have the schedule expire every week.
 - **monthly** to have the schedule expire every month.
 - **at specific date** to have the schedule expire only at a specific date.
 - **never expires** to never have the schedule expire. The same query results will be used every time the project is run.
 - **always expired** to always have the schedule expired. The query results will never be saved and the query will be rerun every time the project is run.
 - **after the visual project is closed** to have the schedule expire after you close the visual project.
4. If you selected **never expires** or **always expired**, click **Finish**.

The **Expiration Schedule Configuration** wizard closes and the new expiration schedule is saved under the **Expiration Schedules** node in the **Project Explorer** view.
5. If you selected any of the other options, click **Next**.

The **Set schedule properties** page of the **Expiration Schedule Configuration** wizard is different depending on your selection.
6. Specify the **Set schedule properties** options and click **Finish**.

The **Expiration Schedule Configuration** wizard closes and the new expiration schedule is saved under the **Expiration Schedules** node in the **Project Explorer** view.

Specifying hourly expiration schedule options

Hourly expiration schedules expire every hour.

About this task

To set the options for an hourly expiration schedule:

Procedure

1. Open the **Set schedule properties** page of the **Expiration Schedule Configuration** wizard.
2. Specify the date that the schedule will begin to expire in the **Begin date** field.
3. Specify at which time on the specified date the schedule will begin to expire in the **Begin time** field.
4. Set the expiration interval by specifying the number of hours and minutes in the **Expiration interval** group of fields.
5. Click **Finish**.

The **Expiration Schedule Configuration** wizard closes and the new expiration schedule is saved under the **Expiration Schedules** node in the **Project Explorer** view.

Specifying daily expiration schedule options

Daily expiration schedules expire on a day to day basis.

About this task

To set the options for a daily expiration schedule:

Procedure

1. Open the **Set schedule properties** page of the **Expiration Schedule Configuration** wizard.
2. Specify the date that the schedule will begin to expire in the **Begin date** list box.
3. Specify at which time on the specified date that the schedule will begin to expire in the **Begin time** list box.
4. Specify an option in the **Data expires** radio group. Valid options are:
 - **daily**: the schedule will expire every day at the specified begin time, starting at the specified begin date.
 - **each**: the schedule will expire after the number of days specified in the **day** list box, at the specified begin time, starting at the specified begin date.
 - **days of week**: the schedule will expire every day of the week that is checked, at the specified begin time, starting at the specified begin date.
5. Click **Finish**.

The **Expiration Schedule Configuration** wizard closes and the new expiration schedule is saved under the **Expiration Schedules** node in the **Project Explorer** view.

Specifying weekly expiration schedule options

Weekly expiration schedules expire on a week to week basis.

About this task

To set the options for a weekly expiration schedule:

Procedure

1. Open the **Set schedule properties** page of the **Expiration Schedule Configuration** wizard.
2. Specify the date that the schedule will begin to expire in the **Begin date** list box.
3. Specify at which time on the specified date that the schedule will begin to expire in the **Begin time** list box.

4. Specify how many weeks from the specified begin date until the schedule expires in the **each week** list box.
5. Specify on which days of the specified week that the schedule will expire in the **days of week** check box group.
6. Click **Finish**.

The **Expiration Schedule Configuration** wizard closes and the new expiration schedule is saved under the **Expiration Schedules** node in the **Project Explorer** view.

Specifying monthly expiration schedule options

Monthly expiration schedules expire on a month to month basis.

About this task

To set the options for a monthly expiration schedule:

Procedure

1. Open the **Set schedule properties** page of the **Expiration Schedule Configuration** wizard.
2. Select the date that the schedule will begin to expire in the **Begin date** list box.
3. Select at which time on the specified date that the schedule will begin to expire in the **Begin time** list box.
4. Specify an option in the **Data expires** radio group. Valid options are:
 - **day** to have the schedule expire on the day of the month specified in the **day** list box.
 - **or** to have the schedule expire at the same relative day of the month. In the first drop-down list, you specify that the schedule will expire on the first, second, third, fourth, or last day that you specify in the second drop-down list.
5. Specify which months the schedule will expire in the **Select months** check box group.
6. Click **Finish**.

The **Expiration Schedule Configuration** wizard closes and the new expiration schedule is saved under the **Expiration Schedules** node in the **Project Explorer** view.

Specifying options for schedules that expire on specific dates

Expiration schedules can be set to expire only on specific dates.

About this task

Procedure

1. Open the **Set schedule properties** page of the **Expiration Schedule Configuration** wizard.
2. Click the first row of the **Date and time** list.

Click the button with the three dots on the right. The **Select date and time** window opens.
3. Use the left and right arrow buttons to select the month and year that the schedule will expire.
4. On the calendar, select the day of the month that the schedule will expire.
5. In the **Time** field, select the time at which the schedule will expire.
6. Click **OK**.

The **Select date and time** window closes and the new date and time is stored in the **Date and time** list.

7. You can repeat these steps to specify multiple dates and times when the schedule will expire.
8. Click **Finish**.

The **Expiration Schedule Configuration** wizard closes and the new expiration schedule is saved under the **Expiration Schedules** node in the **Project Explorer** view.

Working with translation tables

Translation tables enable visual designers to publish a single visual dashboard that can be viewed in multiple languages.

About this task

When content developers create a dashboard, button captions, labels, tooltips and other embedded text is entered directly into the property values of the dashboard objects. Using translation tables, content developers can supply alternate versions of this entered text, to be displayed when the user is in a different locale. For example, a Japanese developer can elect to author a new dashboard in Japanese but then also include translation tables for German and French, each of which will be used when the dashboard is viewed under German and French locales.

Every new dashboard is created with a default translation table. The default translation table provides the collection of supplied text that has been added directly within the property values of the components that make up the dashboard. The contents of the default translation table will be used, unless there is an alternate text value available in a translation table that is more appropriate for the current locale. Consider the following example:

1. A dashboard is authored in Japanese and includes seven text strings.
2. The dashboard developer decides to include a German translation table, providing German translations for three of the seven original text strings. The developer also adds a French translation table, providing translations for all seven strings.
3. When the dashboard is viewed in Japan, all seven original text values are displayed. When viewed in Germany, the three translated strings are used, along with the four original Japanese strings. When viewed in France, the seven French translated strings are used. When viewed in Italy, the Japanese strings are used.

Any number of translation tables can be added to a dashboard project, each of which is associated with a specific language. Although it is possible to directly edit the strings within the translation table editor, the application also provides a means of exporting and re-importing translation tables, allowing content developers to perform the translation using external tools and applications.

Procedure

1. From the **Project Explorer** view, expand the **Globals** node for the dashboard. Right-click the **Translation Tables** node. Select **Insert New Table** from the pop-up menu. The **New Translation Table** wizard opens.
2. Select a language from the **Language** drop-down list. The language that you select will be used as the new translation table name.
Only one translation table can be defined for a given language. If a translation table already exists for a selected language, an error message will appear over the pull-down menu.
3. Click **Finish**.
The **New Translation Table** wizard closes, and the **Edit Translation Table** window opens.
4. The new translation table is populated with the contents of the default translation table. An object's text property value ID is listed in the **ID** cell of the translation table. The text that has been defined for the object's text property value is listed in the **Text** cell of the translation table.
5. Using the check boxes on the left, clear the check box that is associated with each ID that will be excluded from translation.

Only the values for the IDs that remain checked will be translated. Dashboard designers will recognize which values should be translated. Some text values such as company names do not need translating so they should be excluded. Some text property values are not simply strings but calculated (displayed in the text cell with an = sign). If these objects include string values, they should be included for translation. When an object ID is disabled in a translation table, the corresponding value in the default translation table will be used (text values will be in the original

language used to create the dashboard). You can use the **Enable** icon to enable one or more objects if they have been disabled.

6. You can choose to translate the strings within the **Edit Translation Table** window. You can also export the strings for translation to a file. To export the strings to a file, select **Export**.

An operating system specific **Save** window opens.

7. Browse to the directory where you will save the file and click **Save**.

The file name can be different from the language under which the translated table will be saved. Make the file accessible to the translators.

8. The following is an example of how the strings appear in the exported file:

TestDashboard.Level1.Label10.Text=Current Assets. Translators will translate only the actual text in the string and not the syntax of the object ID. This text appears to the right of the equal sign (=). In the example, only **Current Assets** should be translated.

9. When the translation is complete, you will upload the file. From the **Project Explorer**, right-click the translation table that you will be uploading, select **Edit Table** from the pop-up menu. Click **Import**.

An operating system specific **Open** window opens.

10. Navigate to the translated file and click **Open**.

The new changes are shown in the **Text** column of the **Edit Translation Table** window.

11. Click **OK**.

The new translation table is saved and its corresponding language is listed under the **Translation Tables** node for the visual dashboard.

12. In the **Properties** view of the active visual dashboard, specify the **Default language** of the dashboard.

Valid options include:

- **Auto**: The native language of the machine on which QMF for Workstation was installed.
- **Default**: The original language used to create the visual dashboard content.
- If any translation tables have been created for the selected visual dashboard, their corresponding languages will also be listed as default language options.

Switching translation tables in the scene of a dashboard

You use the **Translation Table** global parameter to allow users to change the translation of text values in a dashboard from a scene.

About this task

There might be a time when a dashboard user from a certain locale wants to select a different language in which to view text than the default language. You can use the **Translation Table** global variable to create controls that allow a user to switch from one translation table to another. The following steps describe one common way in which to create a user interface that gives a user this ability:

Procedure

1. Create translation tables for each language that you want to include as an option in your dashboard.
2. From the **Palette** view, select the list box object from the **Controls** palette and click and drag the shape in the editor window.

The **List Box Wizard** opens.

3. Select **No** and click **Next**.

The **Enter option and value list** page opens.

4. Click **Finish**.

The **List Box Wizard** closes and the list box displays in the editor window.

5. In the **Properties** view, select the **OptionList** property and click the **Edit with Expression Designer** button in the toolbar.

The **Expression Designer** opens.

6. Expand the **Translation** folder in the functions tree and double-click the **getAvailableTranslationTables()** function.
7. Click **OK**.
The **Expression Designer** closes and all of the translation table languages that you created are listed in the list box.
8. From the **Palette** view, select the button object from the **Controls** palette and click and drag the shape in the editor window.
9. With the button selected in the editor, open the **Events** view and double-click the **Click** event.
The **[Button Name] Behavior** window opens.
10. Click the **Add New Action** button.
The **Add New Action** window opens.
11. Double-click the **Set values** action.
The **Set values action parameters** page opens.
12. Click the **Add New Entry** button and then click the ellipsis (...) in the **Parameter** cell.
The **Expression Designer** opens.
13. From the **Parameters** folder of the **Globals** node, double-click **Translation Table**.
14. From the **List Box1 properties** folder of the **ListBox1** node, double-click the **Selection** property and click **OK**.
The Expression Designer closes and control returns to the **Add New Action** window.
15. Click **Finish**. The **Set values** action is added. The **Add New Action** window closes.
16. From the **[objectname] Behavior** window, click the **Apply** button to apply all actions that have been added for the event.
17. Click **OK** to close the **[objectname] Behavior** window.
18. Click the **Runtime** tab, select a language from the list box, and click the button.
The next time a process is run in the dashboard (such as running a query) the text values of the scene should display in the selected language.

Profiling visual dashboards

You use profiling to analyze the complexity of visual dashboards scenes and objects they contain dynamically; to measure the data on query running and objects creating and also the time spent on compiling the whole visual project.

About this task

With the help of profiling you can monitor the work of scenes and objects and find the problematic ones. Using this information you can optimize your visual dashboards.

Analyzing the complexity of a scene

You can analyze the complexity of a particular dashboard scene with the help of the **Analyzer** view.

Before you begin

You must have a visual project to analyze and open the **Analyzer** view.

About this task

Analyzer counts the weight of each object lying on a scene. Each object has a number of properties that contain values and expression of different complexity. All default properties are basic units and cannot be subdivided into simpler ones. Each default property has the weight that equals 1. The weight of non-default properties is counted as an expression according to their complexity. In the analysis process you can count the weight of each object and get the overall weight of a scene.



The objects that can run queries or the DataTemplates have incomputable weight. They are highlighted in red in the **Analyzer** view.

The analysis data is represented in a table with the following columns:

- **Elements** column displays a tree structure of the analyzed scene.
- **Weight** column displays a weight of a particular element.
- **Cumulative weight** column displays a weight of a particular parent and the child elements.

To analyze a scene:

Procedure

1. Open the scene you want to analyze in **Canvas**, **Design**, or **Runtime** mode.
2. In the **Analyzer** view click **Analyze** .
3. The analysis of the scene is displayed in the **Analyzer** view as a tree listing all the components and objects of the scene.
4. You can select an object in the view and it will be displayed as selected on the scene and in the **Project Explorer** view.
5. You can specify the values for objects with incomputable weight by clicking the ellipsis (...) in the **Cumulative Weight** column or **Set row count**  button on the view toolbar. This value will be used as an average row amount in the incomputable object.

Profiling a visual dashboard

You can profile a visual dashboard with the help of the **Profiler** view.

Before you begin

You must have a visual project to analyze and open the **Profiler** view.





About this task






The **Profiler** view displays the data on query running, the amount of created objects, and the time spent on creating them and calculating expressions on the visual project. The profiler data is represented in a table with the following columns:

- **Elements** column lists all the scenes with the objects, expressions and queries of a visual project.
- **Count** column shows the number of times the object was created.
- **Time** column displays the time spent on creating an element.
- **Cumulative time** column displays the time spent on creating a parent and its child elements.
- **Percentage** column displays the percentage ratio of the time spent on creating a particular element to the time spent on creating the entire project.

To start the profiling process:

Procedure

1. Click the **Profile**  button on the **Profiler** view toolbar.
2. Open the required visual dashboard or a separate scene in the **Runtime** mode.
3. When all scenes are created and all queries are run click the **Profile**  once again.
The profiler results are displayed in the tree table.
4. You can modify the way the results are displayed by clicking **Tree view**  or **Flat view**  buttons on the **Profiler** view toolbar.

5. You can expand or collapse the tree of elements by clicking **Expand All** () or **Collapse All** () buttons on the **Profiler** view toolbar.
6. To view the SQL text of a query in the project, select the query node in the **Profiler** view and click the **SQL Text** () toolbar button.
7. You can export or import the profiler data by clicking the **Import data** () or **Export data** () toolbar buttons.
8. You can select an object in the view and it will be displayed as selected in the **Project Explorer** view.
9. You can sort the values in the **Profiler** view columns by clicking their headers.

Debugging visual dashboards

You can control the execution of the processes and check for the errors that can exist in visual dashboards using the **Expressions** view. You can check parameter values, object properties, and expression evaluations for the appropriate scene of the visual project in **Runtime** mode.

About this task

To check a scene, do the following steps:

Procedure

1. Open the **Expressions** view. To do this, from the main menu, select **Window > Show View > Other**, expand the **Visual Projects Debugger** node, select **Expressions**, and click **OK**.
2. From the **Expressions** view, click **Watch Global Parameters** to check the values of the global parameters of the visual project for the scene when you run it.
3. Click **Watch Local Parameters** to check local parameters of the scene when you run it.

Note: Global and local parameters specific only for the scene that you run are displayed.
4. You can add an expression that can contain functions, refer to the object properties or scene parameters and watch the evaluation of this expression. Click **Add Expression** to open the **Add Expression** dialog.
5. Type the text of the expression and click **OK** to add this expression to the **Expression** list.

Note: The added expressions are evaluated for all the scenes that you run.
6. To change the existing expression, select it from the **Expression** list and click **Edit Expression**. The **Edit Expression** dialog with the text of the selected expression opens.
7. Edit the text of the expression and click **OK**.

Tip: If you have a scene open in either **Canvas**, **Design**, or **Runtime** mode, the **Expression Designer** window opens by clicking **Add Expression** or **Edit Expression**. You can enter expressions using the expression designer functions.
8. To watch the values of the scene object properties, right-click the object in either **Canvas** mode, **Design** mode, or **Project Explorer** view, and select **Watch Properties** from the menu. The **Select Properties to Watch** dialog opens.
9. Select check boxes corresponding to the object properties that you want to watch and click **OK**. The properties are added to the **Expression** list.
10. Open the scene that you want to check in **Runtime** mode. In the **Value** list of the **Expressions** view, check the global and local parameters and evaluations of the expressions for the current scene.
11. Click **Refresh Expressions** to update the values of the expressions for the active scene.
12. To delete the expression from the **Expression** list and stop watching it, select it from the list and click **Remove Expression**.
13. Click **Clear Expressions** to delete all expressions that are watched from the **Expression** list.

Editing queries of particular objects in visual projects

You can quickly view and edit a query that is used by a particular object in the visual project.

Before you begin

The query used by the object of the visual project must have parameters.

About this task

To open and edit a query of a particular object in a visual project, perform the following steps:

Procedure

1. From the **Project Explorer** view, expand the node of the object for which you want to view the specified query.
2. In the tree of the object, expand the **QueryParameters** node and select the item containing query parameters.
The names of the parameters and their current values are displayed in the **Properties** view. You can change the values of the query parameters.
3. Right-click the selected item with query parameters and select **Open Query** from the menu. The query used by the object is opened in the Visual Query editor. You can modify the query in this editor.

Chapter 16. Working with JavaScript Tables

You can use JavaScript Tables to collect the data from network resources and process it in the form of relational tables.

For example, you need to collect the stock exchange information from a financial portal for a certain period of time and save this information so that you can use it in QMF as a typical relational table.

For this, you create a JavaScript Table, specify its structure, and save it to a JavaScript data source or a virtual data source. In the JavaScript Table editor, you write a script that populates the table with the required data retrieved over HTTP or HTTPS. When you run a query based on the created JavaScript Table, the script retrieves the data and saves it in a table that is used by the query.

If it is necessary, you can use external functions and objects from different modules that are either saved in a repository or located on a web server.

You can run queries based on a JavaScript Table and use them to create quick reports, forecasts, or add them to visual dashboards.

Creating JavaScript Tables

You use the Create JavaScript Table wizard to add a new JavaScript Table and specify its structure.

About this task

To create a JavaScript Table:

Procedure

1. Select **File > New > Other**. In the opened window select **JavaScript Tables** and click **Next**.
The **Create New JavaScript Table** wizard opens.
2. On the **Add New JavaScript Table** page, specify the name for the table you create and select a JavaScript data source or a virtual data source where you want to save it. Click **Next**.
Note: The **Data Source** list contains only virtual data sources from the current repository connection.
3. On the **JavaScript Table Structure** page, define the table structure including the list of columns, their order, value types and required parameters. You can also add comments to any column.
4. Click **Finish** to close the wizard.
The JavaScript Table with the specified name and structure is added to the selected JavaScript data source or virtual data source. After closing the wizard, **JavaScript Table** editor opens.
5. In the **JavaScript Table** editor, you create scripts to populate the JavaScript Table with the required data.
6. When you finish working with script, save the changes and close the editor.

Results

Now you can use the created JavaScript Table as a typical relational table. For example, you can run queries based on the created JavaScript Table, use them to create quick reports, forecasts, or add to visual dashboards.

Writing scripts to populate JavaScript Table

Once you created a JavaScript Table and specified its structure, you need to write a script that will populate the table with the required data.

About this task

You write the script in the **JavaScript Table** editor according to general JavaScript syntax rules. The editor opens automatically after you close the **Create New JavaScript Table** wizard.

To manage the content of a JavaScript Table, you can also use the following special functions of the table global object:

appendData()

Inserts the data specified by a two-dimensional array to the end of the JavaScript Table.

replaceData()

Completely replaces the content of a JavaScript Table with the data specified by a two-dimensional array of the following form: `[[<row 1 column1>, <row 1 column 2>,...], [<row 2 column 1>, <row 2 column 2>,...],...]`.

eraseData()

Removes JavaScript Table records that satisfy the specified SQL condition. If you call `eraseData()` function without arguments or with *null* argument, all table records are removed.

setTag()

Links the JavaScript Table with some additional information specified by a string. This information can be retrieved by the **getTag()** method. To clear the information, use **setTag()** with *null* argument or without arguments.

getTag()

Returns the additional information related to the JavaScript Table.

include()

You can include external code in your JavaScript Table script by using the `include()` method.

To trace HTTP request, response, and record logs, use the `log()` function.

You can also use prompted variables in the script. A user will be prompted to enter variable values when running a query based on the created JavaScript Table. If you use a prompted variable in the script, note the following peculiarities:

- The JavaScript Table cache is ignored.
- When a user works with a JavaScript Table in QMF for WebSphere, all the retrieved data will be available to that user only.

You can work with the preset modules or create custom ones.

When you finish working with the script, save the changes and close the editor.

Procedure

1. If you want to work with the preset modules, add JavaScript to them. For more information about modules, see the *Modules in JavaScript data sources* topic of Administrator Help.
2. To add a user module:
 - a) In the **Repository Explorer** view, right-click the **Modules** node of the data source and select **New > JavaScript Module**.
 - b) In the **New JavaScript Module** wizard, specify the name for the JavaScript Module in the **Name** field and click **Finish** to open the JavaScript Module editor.
3. Type script in the modules and save them.

Important: If you work with preset modules, do not duplicate names of variables in different modules to avoid conflicts during their execution.

JavaScript Table Example

The following example demonstrates how to use JavaScript Tables.

Purpose

You need to collect stock exchange information from a financial portal for a certain period of time and save this information so that you can use it in QMF as a typical relational table. The table must not be updated too often and store information longer than 30 days.

Preparing table structure

Before you begin writing the script, you need to make sure that the specified table structure corresponds to the data that you need to retrieve.

Planning the script structure

According to the purpose the script must retrieve information from server, parse the data to fill the table, save and check the last modification time, and remove outdated information.

```
var lastModificationDate = getTableModificationTime();
if (needUpdate(lastModificationDate))
{
    var raw_data = retrieveRawData(url);
    var data = parseData(raw_data);
    table.appendData(data);
    eraseOutdated();
    setTableModificationTime();
}
else
{
    log("Data is already up to date ");
}
```

The following functions are used:

- **setTableModificationTime()** to attach the last modification time to the table.
- **getTableModificationTime()** to get the last table modification time
- **needUpdate()** to check if update is required depending on the last modification time.
- **retrieveRawData()** to get the required data from the portal.
- **parseData()** to split the received string according to the table structure.
- **eraseOutdated()** to remove old records.
- **log()** to trace the response status and write information to the Output view.

Retrieving data from the Internet

First of all, you need to retrieve the required data from the Internet. For this, **RSBIHttpRequest** class is used:

```
function getStockData(url)
{
    var request = new RSBIHttpRequest();
    request.open('GET',url, false);
    var response = '';
    request.onreadystatechange = function()
    {
        if(request.readyState === request.Complete)
        {
            response = request.responseText;
        }
    }
    request.send(null);
    return response;
}
```

Tip: If you want to use the HTTPS protocol to send HTTP requests for your JavaScript table, you must first configure your Java virtual machine to handle the SSL certificates properly. If you do not want to do the configuration, use the **RSBIHttpRequest.ignoreCertificates()** method. In this case, your JVM is set to ignore the SSL certificates and the connection that is established is not secure.

The **getStockData()** function returns the response string with all required data from the portal.

It is called from the **retrieveRawData()** function that forms an HTTP request and uses **log()** function to trace response information. You can also use prompted variables (the *symbols* variable) in script – user will be prompted to specify the values when running the query. The *&DSQQW_DQ* global variable is used for substitution of double quotes.

```
function retrieveRawData()
{
    var symbols = &DSQQW_DQ&symbols&DSQQW_DQ;
    var url = "http://finance.yahoo.com/d/quotes.csv?s="+symbols+"&f=sn11d1t1ohgdr";
    var raw_data = getStockData(url);
    log(raw_data);
    return raw_data;
}
```

Parsing received data

Secondly, you need to parse the retrieved data according to your purposes so that you can fill the table with the help of **appendData()** or **replaceData()** functions. The easiest way is to split the response string:

```
function parseData(raw_data)
{
    var data = [];
    var rows = raw_data.split('\n');
    for (var i=0; i < rows.length; i++)
    {
        var row = rows[i].trim();
        if (row !== '')
        {
            var cells = row.split(',');
            data.push(cells);
        }
    }
    return data;
}
```

Specifying data update interval

You can attach modification time (as any additional information in string format) to the table with the help of **setTag()** function. The **getTag()** function is used to retrieve this information.

1. Setting the current time (last modification) as a table tag:

```
function setTableModificationTime()
{
    var currentDate = new Date();
    table.setTag(currentDate.getTime());
}
```

2. Retrieving the last modification time:

```
function getTableModificationTime()
{
    var tag = table.getTag();
    if (tag === null)
    {
        var d = new Date();
        d.setDate(-2); // to force update for the first time use
        tag = d.getTime();
    }
    else
    {
        tag = parseInt(tag);
    }
}
```

```

    return tag;
}

```

3. Checking whether update is required:

```

function needUpdate(lastModificationDate)
{
    var yesterday = new Date();
    yesterday.setDate(-1);
    return lastModificationDate <= yesterday.getTime();
}

```

Removing outdated information

You can use the **eraseData()** function to remove outdated information from the table:

```

function eraseOutdated()
{
    var expirationDate = new Date();
    expirationDate.setDate(-30);
    var day = expirationDate.getDay();
    var month = expirationDate.getMonth() + 1;
    var year = expirationDate.getYear() + 1900;
    var strDate = year + '-' + month + '-' + day;
    table.eraseData("Last Trade Date"<' + "'" + strDate + "'");
}

```

Displaying results

When you finish working with the script, save the changes and close the editor.

To run the query, navigate to the saved JavaScript Table in the **Repository Explorer** view and double-click it. Run the opened query.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
|--------|------|-------------------------|-----------------|-----------------|--------|------------|-----------|----------------|-----------|-------|
| Symbol | Name | Last Trade (Price Only) | Last Trade Date | Last Trade Time | Open | Day's High | Day's Low | Dividend/Share | P/E Ratio | |
| 1 | XOM | Exxon Mobil Corpo | 90.58 | 2013-04-02 | <NULL> | 90.68 | 90.92 | 90.28 | 2.28 | 9.36 |
| 2 | GE | General Electric | 23.34 | 2013-04-02 | <NULL> | 23.20 | 23.37 | 23.09 | 0.72 | 17.86 |
| 3 | MSFT | Microsoft Corpora | 28.80 | 2013-04-02 | <NULL> | 28.61 | 28.85 | 28.52 | 0.86 | 15.72 |

Editing JavaScript Tables

You can edit the structure and the script of already created JavaScript Table.

About this task

To edit JavaScript table:

Procedure

1. In the **Repositories** or **Repository Explorer** view, navigate to the required JavaScript Table.
2. To edit the structure of the JavaScript Table, right-click this table and select **Properties** from the popup menu.
 - a) Open the **Structure** page in the **Properties** window.
 - b) Change the table structure including the list of columns, their order, value types and required parameters. You can also add comments to any column.
 - c) To cancel unsaved changes, click **Restore Defaults**.
 - d) To save the updated structure of the JavaScript Table, click **Apply**, and then **OK**.

Note: All the data will be removed from the table.

3. To edit the script of the JavaScript Table, right-click this table and select **Edit Source Script** from the popup menu. In the opened **JavaScript Table** editor, rewrite the script according to your purposes. Once you finished editing the source script, save the JavaScript Table.

Chapter 17. Creating virtual tables

Use virtual tables to process the collected data in the form of relational tables in virtual data sources.

About this task

Virtual tables are stored in virtual data sources. Use them to edit the metadata in a convenient way, specify cache expiration schedules, and work with calculated columns.

To create a virtual table:

Procedure

1. From the main menu, select **File > New > Other > Virtual Table**.
2. In the **New Virtual Table** wizard, select the table, virtual table, query, or forecast that you want to use as the virtual table source.
3. In the **Name** field, enter the name for the virtual table.
4. From the **Data source** list, select the virtual data source to that you want to add the table.
5. Click **Next** to specify the structure of the virtual table or click **Finish** to create a table with default structure and cache settings.

Specifying structures of virtual tables

When the data structure and metadata of a database table are different, use virtual tables in virtual data sources to edit or convert the metadata in a convenient way.

About this task

To specify the virtual table structure:

Procedure

1. On the second page of the **Create New Virtual Table** wizard or on the **Structure** tab of the **Properties** window, use the **Fields** table to specify the structure of the virtual table.
2. To edit a column name, click a cell in the **Name** column and enter a new name.
3. To specify a different data type, click a cell in the **Type** column and select a data type from the list. Available options are the following:
 - **INTEGER** can be used to store signed whole numbers between -2 147 483 648 and 2 147 483 647.
 - **BIGINT** can be used to store large signed whole numbers between -9 223 372 036 854 775 808 and 9 223 372 036 854 775 807.
 - **DOUBLE** can be used to store large numbers with a floating decimal point.
 - **CHAR** is used to store fixed-length string of characters. For example, you can use it to store employee phone number which is generally fixed in length.
 - **VARCHAR** is used to store variable-length string of characters like employee name.
 - **DECIMAL** is used to store the fixed precision and scale numbers. This data type should be used for precise values, such as currency.
 - **DATE** is used to store a date only.
 - **TIME** is used to store a time only.
 - **TIMESTAMP** is used to store both a date and a time.

Note: DATE, TIME, and TIMESTAMP data types support different format options that define how the values are displayed in the result set.

Note: The supported ranges and the precision of values might vary depending on the database you use.

4. In the **Length** field, specify a maximum number of characters for the VARCHAR type.
5. In the **Precision** field, specify a maximum total number of digits for the DECIMAL type.
6. In the **Scale** field, specify a maximum number of:
 - fractional digits for the DECIMAL type
 - digits in the fractional seconds part for the TIMESTAMP or TIMESTAMP WITH TIMEZONE type
7. Optional: In the **Comment** field, enter a comment for the column in the virtual table.
8. To delete a column from the virtual table, click **Remove Column**.
9. To restore a deleted column, click **Add Column** and select the column that you want to restore in the **Add column** window.
10. To add a calculated column to a virtual table, click the **Add Column** button.
 - a) In the **Add column** window, create a conditional expression for a new calculated column in the **Or, enter an expression here** field.
11. To change the position of the columns in the virtual table, use the **Move Up** or **Move Down** buttons.
12. Click **Next** to specify the cache expiration for the virtual table or click **Finish** to close the wizard.

Specifying cache expiration for virtual tables

You can also schedule caching to retrieve table records from cache rather than from the real data source. It makes the work with such tables faster. Specify the cache settings properly to use only relevant data in tables.

About this task

To specify the cache schedule for a virtual table:

Procedure

1. On the third page of the **New Virtual Table** wizard, select the **Use expiration schedules** check box. The schedule expiration settings become available.
2. From the **Default expiration schedule** list, select the schedule that you want to use as the default one for the whole virtual data source.

Note: To add more schedules, consult the QMF administrator.
3. To assign a particular cache expiration schedule to the table, click the **Schedule name** column of the table, the settings of which you want to modify, and select the required schedule from the list of available ones.

Note: If tables contain any substitutional variables, only *Always Expired* expiration schedule is applied to such tables.
4. If your source table requires authorization for accessing it, you must specify the authorization settings for the virtual table. Click the **Manage Authorization** button. The **Manage Authorization** window opens. All the data sources that require authorization are listed there. This authorization data is used for updating the cache.
5. Click the **Login** column for the required data source. The **Edit Data Source Login** dialog opens.
6. Enter the proper values in the **Login** and **Password** fields and click **OK**.
7. When you finish specifying authorization data for all required data sources, click **OK** in the **Manage Authorization** dialog.
8. Click **Finish** to close the wizard.


Invalidating virtual table and JavaScript table cache manually

You can manually invalidate the cache of a virtual or JavaScript table to make sure that you work with the most current data.

About this task

To manually invalidate the cache of a virtual or JavaScript table, complete the following steps:

Procedure

1. From the **Repositories** or **Repository Explorer** view, right-click the virtual or JavaScript table whose cache you want to invalidate and select **Properties** from the menu.
2. From the **Repositories** or **Repository Explorer** view, right-click the virtual or JavaScript table whose cache you want to invalidate and select **Properties** from the menu.
3. In the **Properties** window, select **Cache Settings**.
4. Click  to invalidate the table cache manually.

Note: Make sure that the cache of the table that you work with is not empty or expired. Otherwise, the



button is not available.

Chapter 18. Working with global variables

Global variables are variables that stay active while the current session of QMF is active. This is in contrast to substitution variables that are active only during the execution of an object (query, form, procedure).

For objects that use substitution variables, values are entered via a window when the object is run. After the object runs, the variable and the value that you used no longer exists. For objects that use global variables, the value currently defined for the global variable is used.

There are two types of global variables:

- *System global variables* are pre-loaded with the QMF application. That means their values are re-initialized each time you open the QMF application. System global variables begin with the letters DSQQW, DSQAO, DSQEC, DSQDC, and DSQCP. You cannot add or delete system global variables. However, you can edit the default values for the system global variables.
- *User global variables* are defined by the user. User global variables can be specified with any unique name that does not begin with the letters DSQQW, DSQAO, DSQEC, DSQDC, and DSQCP. User global variables values are not saved when you terminate the QMF session.

Using the **Global Variables** page of the **Preferences** window you can view, delete and edit your user global variables. You can also edit the values of the system global variables.

Specifying user global variables

You use the **Global Variables** page of the **Preferences** window to view, add, delete and edit your user global variables.

About this task

To view, add, edit and delete user defined global variables:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **Global Variables** from the tree. The **Global Variables** page opens.
2. The system and user global variables that are defined for your session are listed in the **Variables** list box. The name of the global variable is listed in the **Name** field. The current value for the global variable is listed in the **Value** field.
3. To add a new user defined global variable, click the **Add** button. The **New variable** window opens. Type the name of the new user global variable in the **Variable name** field. You can type any combination of unique characters that have not already been used. You can not begin a user global variable with the system global variable prefixes of **DSQQW**, **DSQAO**, **DSQEC**, **DSQDC**, and **DSQCP**. Click **OK**. The user global variable is added to the list and will be available for use during the current QMF session. To specify the initial value of the variable, edit the **Value** cell as described in the following step.
4. To edit the value of a user global variable, click in the **Value** field for the global variable that you want to change. Type the new value for the global variable in the **Value** field.
5. To delete a user global variable, select an existing user global variable from the **Variables** list and click the **Remove** button. The user global variable is deleted.
6. Click **Apply** to save your preferences choices.
7. Click **Restore Defaults** to restore the default preference values.
8. Click **OK** to close the **Preferences** dialog.

Editing system global variable values

You use the **Global Variables** page of the **Preferences** dialog to edit any system global variable values.

About this task

To edit system global variable values:

Procedure

1. Select **View > Preferences**. The **Preferences** window opens. Select **Global Variables** from the tree. The **Global Variables** page opens.
2. The system global variables are listed in the **Variables** list by name. System global variable start with the prefix of **DSQQW**, **DSQAO**, **DSQEC**, **DSQDC**, or **DSQCP**.

Restriction: System global variables whose names start with **DSQAO** cannot be edited.

3. The current system global variable values are listed in the **Value** field. To edit the value of a global variable, click in the **Value** field for the system global variable that you want to change. Type the new value for the system global variable in the **Value** field.
4. Click **Restore Defaults** to restore the default values for all the system global variables.
5. Click **Apply** to save your preferences choices.
6. Click **OK** to close the **Preferences** window.

System defined global variables

System defined global variables are pre-loaded with the QMF application. Their values are re-initialized each time you open the application.

The system defined global variables are divided into the following groups:

DSQQW global variables

Global variables whose names begin with DSQQW provide information about the current query environment.

The following DSQQW global variables are currently available:

| Name | Length | Description |
|-------------------|--------|--|
| DSQQW_AUTOMATION | 1 | Indicates whether the application was started as an Automation server. |
| DSQQW_CONNECTIONS | 1 | Controls the use of database server connections while running a procedure. Value can be zero (0) to minimize the number of connections or one (1) to allow a new connection for each RUN QUERY command. Specifying a value of zero (0) can force the distributed product to reset or complete a data object before continuing execution of a procedure. The default value is zero (0). |
| DSQQW_DQ | 1 | The value of a double quote character. This variable can be used in queries and procedures to eliminate the need for the user to enter quotes with a text value. The default value is the double quote character. |

| Name | Length | Description |
|--------------------|--------|--|
| DSQQW_EXP_DT_FRMT | 1 | <p>The format to use when exporting data with the EXPORT DATA command in a procedure. Specify the value of:</p> <ul style="list-style-type: none"> • zero (0) for text format • two (2) for HTML format • three (3) for CSV format • four (4) for IXF format • five (5) for dbase III files • six (6) for Shapefile format • seven (7) for WQML format • eight (8) for XML format • nine (9) for PDF format • ten (10) for XLS format • eleven (11) for XLSX format |
| DSQQW_EXP_OUT_MDE | 1 | <p>The IXF variation to use when exporting data to an IXF file. Value can be zero (0) for System/370 character-mode IXF or one (1) for PC/IXF. The default value is one (1).</p> |
| DSQQW_FST_SV_DATA | 1 | <p>Controls the use of "fast mode" when saving data with the SAVE DATA command in a procedure. Value can be zero (0) to use regular save mode (not fast mode); one (1) to use fast mode with ORDER BY clause(s) stripped; or two (2) to use fast mode with ORDER BY clause(s). The default value is zero (0). For more information on saving data using fast mode see “Saving data using the regular and fast save methods” on page 218.</p> |
| DSQQW_HTML_REFTEXT | 55 | <p>The text that appears in a report when the &REF variable is used. The default value is "Back To".</p> |
| DSQQW_OLD_PF6_MAP | 1 | <p>Choose behavior for the F6 shortcut key.</p> <p>NO: Use default mapping for F6, that is, create new Visual Query when F6 is pressed.</p> <p>YES: Use QMF 10 Mapping for F6, that is, if visual query is not open, create one. If one or more visual queries are open, then switch to last active SQL tab of a Visual Query.</p> |

| Name | Length | Description |
|---------------------------|--------|---|
| DSQQW_ORIENTATION | 0 | The orientation of the application. The value is zero (0) for left-to-right orientation. The value is one (1) for right-to-left orientation. |
| DSQQW_PROC_FAIL_ON_ERROR | 1 | Stops procedure execution if any of the procedure commands fails. A value of zero (0) specifies the procedure will continue. A value of one (1) specifies the procedure will stop. |
| DSQQW_PROC_OUTPUT | | Output file name for a procedure. |
| DSQQW_PROC_WNDWS | 1 | Controls what happens to intermediate result windows created by running a procedure. The value of zero (0) will close all intermediate windows, leaving only the final result window open at the end of the procedure. The value of one (1) will leave all windows open at the end of the procedure. The value of two (2) will close all intermediate windows, and will also close the procedure window if the procedure is run indirectly (run from another procedure or from the command line). The default value is one (1). |
| DSQQW_QUERY_LANG | 1 | Specifies the subtype of query created when a DISPLAY QUERY command is executed but no query object exists. Value can be zero (0) for a query in the SQL view or one (1) for a query in the prompted view. The default value is zero (0). |
| DSQQW_QUERY_PREP | 1 | Specifies whether the query on a RUN command is to be prepared or run. The results of prepared queries are not returned to the user's workstation. Value can be zero (0) to prepare the query, or one (1) to run the query. The default value is one (1). |
| DSQQW_QUERY_PRESERVE_SORT | 1 | Specifies whether the query sorting order defined by a user is saved within the query and used every time the query is run. Value can be zero (0) - not to preserve the sorting order, or one (1) - to preserve the sorting order. The default value is one (1). |

| Name | Length | Description |
|-------------------|--------|--|
| DSQQW_REUSE_OBJS | 1 | Specifies whether existing windows displaying retrieved objects are reused, or if a new window opens every time an object is selected. Value can be zero (0) to always open objects in new windows, or (1) to activate an existing window if the selected object is already open. The default value is one (1). |
| DSQQW_RPT_COPIES | 10 | Specifies the number of copies to print when printing a report with the PRINT REPORT command in a procedure. The default value is one (1). |
| DSQQW_RPT_FONT | 55 | Specifies the font face name to use when printing a report with the PRINT REPORT command in a procedure. The default value is "Monospaced". |
| DSQQW_RPT_FONT_BD | 1 | Specifies the font bold attribute to use when printing a report with the PRINT REPORT command in a procedure. A value of zero (0) specifies not bold and a value of one (1) specifies bold. The default value is zero(0). |
| DSQQW_RPT_FONT_CS | 3 | The character set of the font to use when printing a report with the PRINT REPORT command in a procedure. The default value is zero (0). |
| DSQQW_RPT_FONT_IT | 1 | Specifies the font italic attribute to use when printing a report with the PRINT REPORT command in a procedure. A value of zero (0) specifies not italic and a value of one (1) specifies italic. The default value is zero (0). |
| DSQQW_RPT_FONT_SZ | 2 | Specifies the font point size to use when printing a report with the PRINT REPORT command in a procedure. The default value is ten (10). |
| DSQQW_RPT_LEN_TYP | 1 | Specifies the type of page length that will be used when printing a report with the PRINT REPORT command or exporting a report with the EXPORT REPORT command in a procedure. Value can be zero (0) to automatically fit the length to the printed page, one (1) to specify an explicit number of lines, or two (2) to specify a continuous report with no page breaks. The default value is zero (0). |

| Name | Length | Description |
|-------------------|--------|---|
| DSQQW_RPT_NUM_CHR | 10 | Specifies the number of characters to fit across a printed page when printing a report with the PRINT REPORT command or exporting a report with the EXPORT REPORT command in a procedure. This has an effect only when DSQQW_RPT_WID_TYP is one (1). The default value is eighty (80). |
| DSQQW_RPT_NUM_LNS | 10 | Specifies the number of lines to fit down a printed page when printing a report with the PRINT REPORT command or exporting a report with the EXPORT REPORT command in a procedure. This has an effect only when DSQQW_RPT_LEN_TYP is one (1). The default value is sixty(60). |
| DSQQW_RPT_ORIENT | 1 | The page orientation to use when printing a report with the PRINT REPORT command or exporting a report with the EXPORT REPORT command in a procedure. Value can be zero (0) for portrait or one (1) for landscape. The default value is zero (0). |
| DSQQW_RPT_OUT_TYP | 1 | The format to use when printing a report with the PRINT REPORT command in a procedure. Value can be zero (0) for text or two (2) for HTML. The default value is zero (0). |
| DSQQW_RPT_TD_TYP | 1 | Date format for TD edit code. Value can be zero (0) for ISO format, one (1) for USA format, two (2) for EUR format or three (3) for JIS format. The default value is zero (0). |
| DSQQW_RPT_TT_TYP | 1 | Time format for TT edit code. Value can be zero (0) for ISO format, one (1) for USA format, two (2) for EUR format or three (3) for JIS format. The default value is zero (0). |
| DSQQW_RPT_USE_PS | 1 | Specifies what page formatting options (page length, page width, etc.) to use when printing a report with the PRINT REPORT command in a procedure. Value can be zero (0) to use the values specified on the PRINT REPORT command or in global variables, or one (1) to use the values specified in the form's page setup. The default value is one (1). |

| Name | Length | Description |
|-------------------|---------------|--|
| DSQQW_RPT_WID_TYP | 1 | Specifies the type of page width when printing a report with the PRINT REPORT command in a procedure. Value can be zero (0) to automatically fit the width to the printed page, one (1) to specify an explicit number of characters or two (2) to specify a continuous line. The default value is zero (0). |
| DSQQW_SHOW_QUERY | 1 | Specifies which view of a query to display when a SHOW QUERY command is issued from a procedure. Value can be zero (0) for SQL or prompted view, one (1) for results view, two (2) for build view, or three (3) for layout view. The default value is zero (0). |
| DSQQW_SQ | 1 | The value of a single quote character. This variable can be used in queries and procedures to eliminate the need for the user to enter quotes with a text value. The default value is a single quote character ("). |
| DSQQW_SV_DATA_C_S | 10 | The number of rows to insert before committing the unit of work when saving data with a SAVE DATA command in a procedure. Value can be zero (0) for all of the rows or an explicit number of rows. The default value is zero (0). |
| DSQQW_SV_DATA_T_M | 1 | Specifies how source and target column data types are matched when using the SAVE DATA command. The value can be zero (0) to require exact data type matches, one (1) to allow data type conversions with no possible data loss, or two (2) to allow all data type conversions that are supported by the database. The default value is one (1). |
| DSQQW_UEDIT_JAR | 55 | The name of the JAVA archive file that contains user edit routines. |

DSQAO global variables

Global variables whose names begin with DSQAO provide information about the current state of the query session.

The following DSQAO global variables are available:

| Name | Length | Description |
|-------------|---------------|--|
| DSQAO_BATCH | 1 | Batch or interactive mode. Value can be one (1) for an interactive session or two (2) for a batch session. See the BATCH command line parameter. |

| Name | Length | Description |
|----------------------|--------|--|
| DSQAO_CONNECT_ID | 8 | The user ID that is used to connect to the current database. |
| DSQAO_CURSOR_OPEN | 1 | The status of the current query object's database cursor. Value can be one (1) if the cursor is open or two (2) if the cursor is closed. |
| DSQAO_DBCS | 1 | DBCS support status. Value can be one (1) if DBCS support is present or two (2) if DBCS support is not present. |
| DSQAO_HOME_WORKSPACE | 128 | <p>The current repository user's home workspace key, if the workspace exists. Valid values are:</p> <ul style="list-style-type: none"> • <code>rsbi:/.workspaces/<user name></code> This is the value if the user connected to a secured repository connection and if the <code>rsbi:/.workspaces/<user name></code> object is viewable by the current user in the repository. <code><user name></code> is the login name of the repository user. • <code>rsbi:/.workspaces</code> This is the value if the user connected to a repository connection without security, or the <code>rsbi:/.workspaces/<user name></code> is not viewable by the user, or it does not exist. • blank This is value if the previous situations do not exist. For example, if the user is not connected to any repository connection. <p>Note: Workspace operations such as creating, deleting, and renaming performed by the current user affect the Global Variable value. Additionally, such operations performed by other users might also affect the value.</p> |
| DSQAO_NLF_LANG | 1 | National language of session. Value is "E" for the English language. |
| DSQAO_NUM_FETCHED | 0 | The number of rows fetched by the current query object. |
| DSQAO_OBJ_NAME | 18 | The name of the current query, form, or procedure object. If there is no current object, the value is blank. |
| DSQAO_OBJ_OWNER | 8 | The owner of the current query, form, or procedure object. If there is no current object, the value is blank. |
| DSQAO_REP_USER | 8 | The user name that is used to connect to the current repository. |
| DSQAO_RSBI_RELEASE | 2 | Numeric release number of the application. |
| DSQAO_RSBI_VER_RLS | 10 | External version and release number for the application. |

| Name | Length | Description |
|-------------------|--------|---|
| DSQAO_QUERY_MODEL | 1 | Model of the current query object. Value can be one (1) for relational. |
| DSQAO_QRY_SUBTYPE | 1 | Subtype of the current query object. Value can be one (1) for SQL queries or three (3) for queries in a prompted view. |
| DSQAO_SYSTEM_ID | 1 | Current [®] operating system. Values can be one of the following: <ul style="list-style-type: none"> • 8 - Windows NT and above • 9 - Linux • 10 - HP-UX • 11 - AIX • 12 - Solaris • 13 - iSeries • 14 - z/OS[®] |

DSQEC global variables

Global variables whose names begin with DSQEC control how commands and procedures are executed.

The following DSQEC global variables are available:

| Name | Length | Description |
|-------------------|--------|---|
| DSQEC_CURR_FOLDER | 128 | Use the variable to group links to objects in folders in QMF Catalog. Its value is used as a default one for the FOLDER parameter in the LIST, SAVE, and ERASE commands for Db2 databases. It can be up to 128 characters long. The variable is blank by default, no folder is used for LIST, SAVE, and ERASE commands. |
| DSQEC_EXTND_STG | 31 | Specifies the number of megabytes of extended storage that the application will acquire on each request to the extended storage manager when spilling data to extended storage in QMF for TSO. When a user performs an operation that requires extended storage, the application issues repeated requests to the extended storage manager for the specified amount until the operation is complete or extended storage is exhausted. When setting this global variable, consider the average size of DATA objects with which your users work. If the average size is very large and you set the DSQEC_EXTND_STG variable too low, the application must issue many calls to the extended storage manager to complete the DATA object, which could affect overall performance. Values can be from 1 to 1000. The default value is 25, indicating that the application requests 25MB of storage on each request. |
| DSQEC_FORM_LANG | 1 | Defines the default NLF language in which a form will be saved or exported. Value can be zero (0) for the presiding NLF language or one (1) for English. The default value is one (1). |

| Name | Length | Description |
|-------------------|--------|--|
| DSQEC_LAST_RUN | 1 | <p>Specifies the set of commands that cause the LAST_USED column of the Q.OBJECT_DIRECTORY table to be updated. Possible values are:</p> <ul style="list-style-type: none"> • 0 - Last used is updated on any activity. • 1 - Last used is updated when RUN, SAVE, or IMPORT commands are performed. |
| DSQEC_LIST_OWNER | 128 | <p>Provides the default value for the OWNER parameter of the LIST command. Specify an authorization ID up to 128 characters long. This variable is blank by default, resulting in a list of objects owned by the current authorization ID. You can use selection symbols in the variable value. Use an underscore (_) in place of a single character and a percent sign (%) in place of zero or more characters. For example, after you issue the following command, followed by a LIST command, the application lists only objects that are owned by user IDs that begin with the characters RO:</p> <pre>SET GLOBAL (DSQEC_LIST_OWNER=RO%</pre> <p>The following command sets the default owner to any user IDs that begin with I, have any character in the second position, and any characters in the remaining positions:</p> <pre>SET GLOBAL (DSQEC_LIST_OWNER=I_%</pre> |
| DSQEC_NLFCMD_LANG | 1 | <p>Defines the expected NLF language for commands in procedures. Value can be zero (0) for the presiding NLF language or one (1) for English. The default value is zero (0).</p> |
| DSQEC_RESET_RPT | 1 | <p>Determines whether a user will be prompted when an incomplete data object that will affect performance is encountered. Value can be zero (0) to complete the data object without prompting, one (1) to prompt the user asking whether the data object should be completed, or two (2) to reset the data object without prompting.</p> |
| DSQEC_RUN_MQ | 1 | <p>Specifies whether the RUN QUERY command supports multiple statements in an SQL query. Possible values are:</p> <ul style="list-style-type: none"> • 0 - Multiple SQL statements are not supported. <p>If you set this variable to 0 and run an SQL query that contains multiple statements, the application ignores all statements after the first semicolon.</p> <ul style="list-style-type: none"> • 1 - Multiple SQL statements are supported. <p>This is the default value.</p> <p>A semicolon can be placed at the end of each statement.</p> <p>You can substitute the semicolon with any other character by using SET STATEMENT DELIMITER comment at the beginning of the SQL text. For example, the following example is a valid use of SQL with multiple statements:</p> <pre>--SET STATEMENT DELIMITER="!" select * from q.staff! select * from q.org</pre> |

| Name | Length | Description |
|-------------------|--------|---|
| DSQEC_SAV_ACCELDB | 128 | Specifies the name of the accelerator database that you want to use to save your tables. |
| DSQEC_SHARE | 1 | Specifies the default value for whether a saved object will be shared with other users. Value can be zero (0) to not share the object or one (1) to share the object. |
| DSQEC_SP_RS_NUM | 1 | Specifies the number of the result set that will be displayed for a stored procedure. The default result set number is minus one (-1). |
| DSQEC_SQLQRYSZ_2M | 1 | Controls whether SQL queries greater than 32,767 bytes (32 KB) in length are supported by the RUN QUERY command. Possible values are: <ul style="list-style-type: none"> • 0 - SQL queries directed to DB2 for iSeries or Linux and UNIX, and Windows databases are limited to 32,767 bytes (32 KB). This is the default. • 1 - SQL queries can be greater than 32 KB. that the version of the database to which the RUN QUERY command is directed supports queries of this size. SQL queries directed to DB2 for VM and VSE are limited to 8 KB. |

DSQDC global variables

Global variables whose names begin with DSQDC control how information is displayed.

The following DSQDC global variables are available:

| Name | Length | Description |
|-------------------|--------|---|
| DSQDC_COL_LABELS | 1 | Specifies whether column headings will be column names or database labels in Classic Reports. Value can be zero (0) to specify that column headings will be column names or one (1) to specify that column headings will be database labels. The default value is one (1). |
| DSQDC_CURRENCY | 18 | Defines the custom currency symbol to use when the DC edit code is specified. |
| DSQDC_DISPLAY_RPT | 1 | Specifies whether a report is displayed after a RUN QUERY command in a procedure. Value can be zero (0) to not display a report or one (1) to automatically display a report with the default form. The default value is zero (0). |
| DSQDC_POS_SQLCODE | 1 | Specifies what happens when a positive SQL code is returned from the database. Possible values are: <ul style="list-style-type: none"> • 0 - Neither log the message or display the message text. • 1 - Log the message associated with the SQL code. • 2 - Display the online help that is associated with the SQL code. |

DSQCP global variables

Global variables whose names begin with DSQCP control the operation of the table editor.

The following DSQCP global variables are available:

| Name | Length | Description |
|--------------|--------|---|
| DSQCP_TEDFLT | 1 | Defines the reserved character used in the Table Editor to indicate a default value for a column. The default value is "+". This variable can also be set in the Enter default values as field found in the Preferences window. |
| DSQCP_TENULL | 1 | Defines the reserved character used in the Table Editor to indicate a null value for a column. The default value is "-". This variable can also be set in the Enter null values as field found in the Preferences window. |

Chapter 19. Working with edit codes

Edit codes specify how to format the data that will be in a specific column of a report.

A default edit code that is based on the data type of the column is automatically assigned to each column in a report. You can change the default edit code for a column in a report from the **Form** dialog. The following types of edit codes are available:

Edit codes for character data

You use character data edit codes to format text fields in reports.

The following character data edit codes are available:

| Edit Code | Format | Notes |
|-----------|---|---|
| C | Display character data. | No special formatting. |
| CW | Display character data with wrapping based on column width. | No special formatting unless the value can not fit on one line in the column. In that case, as much data as possible is placed within the column, and then additional data is wrapped to subsequent lines in the column. |
| CT | Display character data with wrapping based on the text in the column. | No special formatting unless the value can not fit on one line in the column. In that case, as much data as possible is placed on a line within the column, until a blank is found within the text that triggers the wrapping of additional data to subsequent lines in the column. If the string of text is too long to fit in the column and does not contain a blank, the data is wrapped by width until a blank is found. |
| CDx | Display character data with wrapping based on the specified delimiter. | No special formatting unless the value can not fit on one line in the column. In that case, a new line of data begins in the column each time a special delimiter is encountered in the text. If the string of text is too long to fit in the column and does not contain a delimiter, the data is wrapped by width until a delimiter is found. The delimiter specified by "x" can be any single character, including a blank. The delimiter character does not appear in the report. |
| X | Formats data as a series of hexadecimal characters. | No special formatting applied. |
| XW | Formats data as a series of hexadecimal characters with wrapping based on column width. | Columns are wrapped according to the rules specified for the CW edit code. |
| B | Formats data as a series of zeros and ones. | No special formatting applied. |

| Edit Code | Format | Notes |
|------------------|---|---|
| BW | Formats data as a series of zeros and ones with wrapping based on column width. | Columns are wrapped according to the rules specified for the CW code. |

Edit codes for date data

You use date data edit codes to format fields containing date information in reports.

The letter "x" in the date edit codes represents the character to be used as the delimiter in the date value. The value of "x" can be any special character, including a blank, except a letter or a number. The following date data edit codes are available:

| Edit Code | Format | Notes |
|------------------|-----------------|--|
| TDYx | YYYYxMMxDD | Year First |
| TDMx | MMxDDxYYYY | Month first |
| TDDx | DDxMMxYYYY | Day first |
| TDYAx | YYxMMxDD | Two digit year first. |
| TDMAx | MMxDDxYY | Month first and two digit year. |
| TDDAx | DDxMMxYY | Day first and two digit year. |
| TDL | Locally defined | Formats date according to the format specified as the default at the database server requesting data. |
| TDT | Locally defined | Edit codes that appear on result set reports; that is, reports generated from a stored procedure CALL. They would be used on time or date data if the data was found not to be in ISO format. If these edit codes are found on column data, then the edit code cannot be changed for that column. Also, the report object cannot be exported if this edit code is present in the form. |

Edit codes for graphic data

You use graphic data edit codes to format fields containing graphic or pure DBCS information in the reports.

The following graphic data edit codes are available:

| Edit Code | Format | Notes |
|------------------|---|--|
| G | Display graphic data. | No special formatting. |
| GW | Display graphic data with wrapping based on column width. | No special formatting unless the value can not fit on one line in the column. In that case, as much data as possible is placed within the column, and then additional data is wrapped to subsequent lines in the column. |

Edit codes for numeric data

You use numeric data edit codes to format fields containing numeric information in reports.

The letters "nn" in the numeric data edit codes represents a number between 0 and 99. This number determines how many places to allow after the decimal point. The following numeric data edit codes are available:

| Edit Code | Format | Notes |
|------------------|--|---|
| E | Displays numbers in scientific notation. | Up to 17 significant digits, or up to 34 significant digits when editing extended floating point data, are shown even if the width of the column can accommodate more. Used as the default form for columns with data type FLOAT. |
| EZ | Displays numbers in scientific notation with zero values in the column suppressed. | Up to 17 significant digits, or up to 34 significant digits when editing extended floating point data, are shown even if the width of the column can accommodate more. |
| Dnn | Displays numbers in decimal notation formatted with a negative sign, thousands separator, and currency symbol. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |
| DZnn | Displays numbers in decimal notation formatted with a negative sign, thousands separator, currency symbol and any zero values in the column suppressed. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |
| DCnn | Displays numbers in decimal notation formatted with a negative sign, thousands separator, and a user-defined currency symbol. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. The currency symbol that will be used instead of the standard currency symbol is defined using the global variable DSQDC_CURRENCY. |
| DZCnn | Displays numbers in decimal notation formatted with a negative sign, thousands separator, a user-defined currency symbol and any zero values in the column suppressed. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. The currency symbol that will be used instead of the standard currency symbol is defined using the global variable DSQDC_CURRENCY. If both edit code options "Z" and "C" are used, "C" must follow "Z". |
| Inn | Displays numbers in decimal notation formatted with leading zeros displayed and a negative sign. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |

| Edit Code | Format | Notes |
|------------------|--|---|
| IZnn | Displays numbers in decimal notation formatted with leading zeros displayed, a negative sign and any zero values in the column suppressed. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |
| Jnn | Displays numbers in decimal notation formatted with leading zeros displayed. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |
| JZnn | Displays numbers in decimal notation formatted with leading zeros displayed and any zero values in the column suppressed. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |
| Knn | Displays numbers in decimal notation formatted with a negative sign and a thousands separator. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |
| KZnn | Displays numbers in decimal notation formatted with a negative sign, a thousands separator and any zero values in the column suppressed. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |
| Lnn | Displays numbers in decimal notation formatted with a negative sign. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |
| LZnn | Displays numbers in decimal notation formatted with a negative sign and any zero values in the column suppressed. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |
| Pnn | Displays numbers in decimal notation formatted with a negative sign, thousands separator, and a percent sign. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |

| Edit Code | Format | Notes |
|------------------|--|---|
| PZnn | Displays numbers in decimal notation formatted with a negative sign, thousands separator, a percent sign and any zero values in the column suppressed. | The value "nn" is a number (0 to 99) that represents how many places to allow for the decimal point. Numbers with more places after the decimal are rounded and numbers with fewer places are padded. |

Edit codes for time data

You use time data edit codes to format fields containing time information in reports.

The letter "x" in the time data edit code represents the character that will be used as the delimiter in the time value. The value of "x" can be any character, including a space, except a letter or a number. The following time data edit codes are available:

| Edit Code | Format | Notes |
|------------------|----------------------|--|
| TTSx | HHxMMxSS | 24 hour clock, including seconds |
| TTCx | HHxMMxSS | 12 hour clock, including seconds |
| TTAx | HHxMM | Abbreviated 24 hour clock (no seconds) |
| TTAN | HHMM | Abbreviated 24 hour clock (no seconds) without any delimiter between the time values. |
| TTUx | HHxMM PM HHxMM AM | USA format |
| TTL | Locally defined. | Formats time data according to the format specified as the local default at the database server requesting data. |
| TTT | Locally defined | Edit codes that appear on result set reports; that is, reports generated from a stored procedure CALL. They would be used on time or date data if the data was found not to be in ISO format. If these edit codes are found on column data, then the edit code cannot be changed for that column. Also, the report object cannot be exported if this edit code is present in the form. |

Edit codes for timestamp data

You use timestamp data edit codes to format fields containing timestamp information in reports.

The following timestamp edit codes are available:

| Edit Code | Format | Notes |
|-----------|---|--|
| TSI | yyyy-mm-dd-hh.mm.ss.nnnnnnnnnnn | Where: <ul style="list-style-type: none"> • yyyy is the four digit year • mm is the two digit month • dd is the two digit day • hh is the two digit hour • mm is the two digit minute • ss is the two digit second • nnnnnnnnnnn is the twelve digit fractional seconds |
| TSZ | yyyy-mm-dd-hh.mm.ss.nnnnnnnnnnn ±th:tm | Where: <ul style="list-style-type: none"> • yyyy is the four digit year • mm is the two digit month • dd is the two digit day • hh is the two digit hour • mm is the two digit minute • ss is the two digit second • nnnnnnnnnnn is the twelve digit fractional seconds • ±th is the two-digit value representing the time zone hour, shown as an offset relative to UTC • tm is the two-digit value representing the time zone minutes between 0 and 59 <p>Note: To specify UTC, you can either specify a time zone of -24:00 or +24:00 or replace the time zone offset and its sign with an uppercase Z.</p> |

User defined edit codes

You can use user defined edit codes Uxxxx and Vxxxx for special purposes.

The value "xxxx" can be any combination of characters, excluding embedded blanks.

The following user edit codes are predefined:

| Edit Code | Format | Notes |
|-----------|----------------|--------------------------------|
| VSSN | xxx-xx-xxxx | Social security number format. |
| VTEL | (xxx) xxx-xxxx | Telephone number format. |
| VTEL2 | xxx.xxx.xxxx | Telephone number format. |
| VZIP | xxxxx-xxxx | Zip code format. |

Implementing custom edit codes for classic reports

Users can implement custom edit codes that can be used when generating classic reports.

About this task

To implement custom edit codes:

Procedure

1. Create a new Java project and ensure that the Java build path contains the QMF API module located under the API installation location. For example: `productinstalldirectory\API\ibm`. In order for this library to be available, the API component must be selected during the product installation.
2. For each custom edit code, create a class which implements the `FormCustomEditCode` interface.
3. Implement the following methods in this class:
 - Constructor must be without parameters or it can be omitted.
 - The `init(int iColumnWidth, int iColumnAlignment, String strExtParams)` method parameters are defined as follows:
 - `iColumnWidth` - the width in characters of a column with the edit code.
 - `iColumnAlignment` - the alignment of the column. The values that can be specified are `ALIGN_LEFT`, `ALIGN_RIGHT`, `ALIGN_CENTER`, `ALIGN_DEFAULT`.
 - `strExtParams` - typically it is a full string that was specified in the column's "Edit" parameter.
 - The `register()` method will return registration information for this custom edit code.
 - The `format(Object obj)` method formats the specified object as an array of strings. Every string in this array should be of `iColumnWidth` width and all alignment should be performed using space characters. You can use the `StringUtils.getWidth(char, Locale)` function to obtain a specified character's width.

Note: The width of a string is not necessarily equal to the number of characters in it, because some characters, such as BiDi management characters have a width equal to 0; while some characters such as DBCS characters can have a width equal to 2.

 - `destroy()` - This method can be empty.
4. Generate a JAR file for this Java project and place it into the "user.home" directory. For example in Windows XP this directory would be: `C:\Documents and Settings\username\Application Data\IBM\QMF for Workstation`.
5. Start QMF for Workstation. Select **View > Preferences > Global Variables**. From the **Global Variable** window specify a class name of the custom edit code for the `DSQQW_RSUEDIT_JAR` global variable. For example:

```
com.ibm.custom.editcode.sample.VCAN1
```

If you have several custom edit codes you must specify a class name for each separating each with a semi-colon. For example:

```
com.ibm.custom.editcode.sample.VCAN1;com.ibm.custom.editcode.sample.VCAN2
```

6. Re-start the application.

Sample implementation for custom edit codes

The following sample class must be implemented to define each custom edit code. In the following sample `VCAN1` is the name of the custom edit code.

The implemented class must have public constructor without parameters. Do not add any operations in constructor, use the `init()` method.

```
package com.ibm.custom.editcode.sample;
```

```

import java.util.Date;
import java.util.Locale;

import com.ibm.bi.core.util.StringUtils;
import com.ibm.bi.qmf.core.model.form.FormCustomEditCode;
import com.ibm.bi.qmf.core.model.form.FormCustomEditCodeInfo;

public class VCAN1 implements FormCustomEditCode
{
    private int m_iWidth;
    private int m_iAlignment;
    private String m_strExtParams;

    /**This method is called after all operations to destroy structures
    */
    public void destroy()
    {
    }

    /**This method is called to format data.
    *
    * @param obj is an object to format. Parameter class can be String,
    * java.lang.Number, or java.util.Date depending on
    * data read from database or returned by NetRexx calculation.
    * You can restrict the passing of one or more of these types during
    * the registration of the custom edit code.
    * NaN, infinite (positive or negative), and null values can be passed
    * to this method if it was requested at registration of custom edit code.
    * The values are passed as strings in standard form. For example:
    * "DSQNULL" (for null values),
    * "DSQUNDEF" (for NaN values),
    * "DSQNOINS" (for LOB values).
    *
    * @return returns formatted object as a string array. This array is a set of
    * text lines that will be output as a column value. Be sure that returned
    * lines have appropriate width. The width must be equal to iColumnWidth
    * parameter in the init() method.
    *
    * @exception IllegalArgumentException (or any another RuntimeException) can
    * be thrown to show that the data type is incompatible with this edit code.
    * An exception can also be thrown for other reasons when the data
    * cannot be formatted by this edit code. In the case of an exception,
    * a "dummy string" (string of spaces) can be output as value
    * of this edit code.
    */
    public String[] format(Object obj) throws IllegalArgumentException
    {
        if(isNull(obj))
        {
            return formatNull();
        }
        else if(isNaN(obj))
        {
            return formatNaN();
        }
        else if(obj instanceof Number)
        {
            return format((Number)obj);
        }
        else if(obj instanceof Date)
        {
            return format((Date)obj);
        }
        else if(obj instanceof String)
        {
            return format((String)obj);
        }
        //Never raised
        throw new IllegalArgumentException();
    }

    private String[] format(Number num)
    {
        return align(num.toString());
    }
    private String[] format(String str)
    {
        return align(str);
    }
    private String[] format(Date date)

```

```

{
    return align(date.toString());
}
private String[] formatNull()
{
    return align("<NULL>");
}
private String[] formatNaN()
{
    return align("<NaN>");
}

/**
 * This is a simple helper routine to determine if the object passed
 * to FormCustomEditCode.format() function represents null.
 */
public static Boolean isNull(Object obj)
{
    return "DSQNULL".equals(obj);
}

/**
 * This is a simple helper routine to determine if the object passed
 * to FormCustomEditCode.format() function represents NaN.
 */
public static Boolean isNaN(Object obj)
{
    return "DSQUNDEF".equals(obj);
}

/**
 * This is a simple helper routine to determine if the object passed
 * to FormCustomEditCode.format() function represents LOB.
 */
public static Boolean isLob(Object obj)
{
    return "DSQNOINS".equals(obj);
}

/**
 * This is a simple helper routine to determine if the object passed
 * to FormCustomEditCode.format() function represents special
 * objects like null, NaN or LOB.
 */
public static Boolean isSpecial(Object obj)
{
    return isNull(obj) || isNaN(obj) || isLob(obj);
}

/**You should implement this method to store the
 * column width and requested column alignment.
 * This method is called before
 * any format operation.
 * */
public void init(int iColumnWidth, int iColumnAlignment, String strExtParams)
{
    m_iWidth = iColumnWidth;
    m_iAlignment = iColumnAlignment;
    m_strExtParams = strExtParams;
}

private String[] align(String strLine)
{
    return align(strLine, m_iAlignment, m_iWidth);
}

/**
 * Simple alignment routine.
 * This routine operates on a single line of text.
 */
public static String[] align(String strLine, int alignment, int iWidth)
{
    switch (alignment)
    {
        case ALIGN_CENTER:
        {
            return new String[] {alignStringToCenter(strLine, iWidth)};
        }

        case ALIGN_RIGHT:
        {
            return new String[] {alignStringToRight(strLine, iWidth)};
        }
    }
}

```

```

    }

    case FormCustomEditCode.ALIGN_LEFT:
    case FormCustomEditCode.ALIGN_DEFAULT:
    default:
    {
        return new String[] {alignStringToLeft(strLine, iWidth)};
    }
}

/**
 * Formats string, adding ' ' character to the right
 * and to the left so the resulting length is iWidth
 */
private static String alignStringToCenter(String strSource, int iWidth)
{
    StringBuffer sbSource = new StringBuffer();
    int iLength = ensureStringToFit(strSource, iWidth, sbSource);

    int iAddLen = iWidth - iLength;
    if (iAddLen >= 0)
    {
        return fillSpaces(sbSource, iAddLen / 2, iAddLen - (iAddLen / 2));
    }
    return sbSource.toString();
}

/**
 * Formats string, adding ' ' character to the left so
 * the resulting length is iWidth
 */
private static String alignStringToRight(String strSource, int iWidth)
{
    StringBuffer sbSource = new StringBuffer();
    int iLength = ensureStringToFit(strSource, iWidth, sbSource);

    int iAddLen = iWidth - iLength;
    if (iAddLen >= 0)
    {
        return fillSpaces(sbSource, iAddLen, 0);
    }
    return sbSource.toString();
}

/**
 * Formats string, adding ' ' character to the right so
 * the resulting length is iWidth
 */
private static String alignStringToLeft(String strSource, int iWidth)
{
    StringBuffer sbSource = new StringBuffer();
    int iLength = ensureStringToFit(strSource, iWidth, sbSource);

    int iAddLen = iWidth - iLength;
    if (iAddLen >= 0)
    {
        return fillSpaces(sbSource, 0, iAddLen);
    }
    return sbSource.toString();
}

/**
 * Verify that a string fits into specified width
 */
private static int ensureStringToFit(String strSource,
    int iWidth, StringBuffer sbResult)
{
    Locale loc = Locale.getDefault();
    char[] carrSource = strSource.toCharArray();
    int iResultWidth = 0;
    for (int i = 0; i < carrSource.length; i++)
    {
        char c = carrSource[i];
        //Calculate character width. 0 for BiDi managing characters,
        // 2 for DBCS characters, 1 for others
        int iCharWidth = StringUtils.getCharacterWidth(c, loc);
        if (iResultWidth + iCharWidth > iWidth)
        {
            //Specified string does not fit into specified Width.
            //So, fill result with asterisks
            sbResult.setLength(0);
        }
    }
}

```

```

        sbResult.append(StringUtils.duplicateCharacter('*', iWidth));
        return iWidth;
    }
    iResultWidth += iCharWidth;
    sbResult.append(c);
}
return iResultWidth;
}

/**
 * Append specified number of space characters to the right and to the left
 */
private static String fillSpaces(StringBuffer sbResult, int iNumLeftSpaces,
int iNumRightSpaces)
{
    if (iNumLeftSpaces > 0)
    {
        sbResult.insert(0, StringUtils.duplicateCharacter(' ', iNumLeftSpaces));
    }
    if (iNumRightSpaces > 0)
    {
        sbResult.append(StringUtils.duplicateCharacter(' ', iNumRightSpaces));
    }
    return sbResult.toString();
}

/**Implement this method to register the custom edit code.
 * This method is called once for this class.
 * (This method can be re-called if registration is
 * requested again by the user application, for example changing registration
 * file)
 */
public FormCustomEditCodeInfo register()
{
    return new FormCustomEditCodeInfo(
        "VCAN1",
        "",
        FormCustomEditCodeInfo.ACCEPT_SPECIAL_VALUES |
        FormCustomEditCodeInfo.ACCEPT_NUMBERS |
        FormCustomEditCodeInfo.ACCEPT_DATES |
        FormCustomEditCodeInfo.ACCEPT_STRINGS |
        FormCustomEditCodeInfo.AUTOCONVERT_TO_STRING);
}
}
}

```

Edit codes for metadata

You can use the metadata edit code **M** to display in a report the descriptive data for a column rather than the actual data.

| Edit Code | Format | Notes |
|-----------|---|---|
| M | Displays in character format a column's metadata. | The metadata for a column is found in the Descriptor Area (DA) and consists of the type and the length of the data that will be included in the column. If a column with the edit code M is null, a null indicator is displayed rather than the metadata. If the column size is less than the amount needed to display the metadata, the metadata is truncated in order to fit into the column space. |

Chapter 20. Working with Analytics objects

You can open, export, and print QMF Analytics for TSO objects that you created in QMF for TSO and CICS and saved to the QMF Catalog. You can find these objects in the Analytics node of the QMF Catalog in QMF for Workstation. These Analytics objects are opened as charts in QMF for Workstation.

Before you begin

This feature is available only for Db2 for z/OS database.

About this task

You can open, export, or print the following QMF Analytics for TSO objects in QMF for Workstation:

- histogram
- mixed
- pie
- plot
- tower

Opening Analytics objects from QMF Catalog

You can open Analytics objects that you created in QMF for TSO and CICS and saved to the QMF Catalog in QMF for Workstation.

Before you begin

You must have a QMF Analytics for TSO object saved to the QMF Catalog.

About this task

You can preliminary open a source query or table for Analytics objects or you can add it later after you open an object. See the following topics for more information:

Opening Analytics objects with a source

You can open a source query or table for Analytics objects.

About this task

To open an Analytics object with a preliminary opened source query or table:

Procedure

1. Open a source query or table.
2. Select **File > Open From > QMF Catalog** and open the required object from the QMF catalog (See [Opening QMF catalog objects](#)).

The object results are displayed in the editor area.

Results

You can export or print the opened Analytics object.

Opening Analytics objects without a source

You can open Analytics objects without a source query or table and add it later.

About this task

To open an Analytics object without a preliminary opened source query or table

Procedure

1. Select **File > Open From > QMF Catalog** and open the required object from the QMF catalog (See [Opening QMF catalog objects](#)).
2. In the **Select Columns** dialog, click ellipsis (...) to specify a query or table, which you want to use as a source for the chart. The **Data Source Object** dialog opens.
3. In the **Data Source Object** dialog select a directory, from which you want to open a query or table:
 - a) If you select **From file** option, enter the required query or table in the **Name** field using the **Browse** button.
 - b) If you select **From data** source option, click the **List Objects** button.
The **Object List** dialog opens. For detailed information about working with the Object List dialog, see steps 3-8 in the [Listing data source objects](#) topic.
 - c) If you select **From repository** option, the current repository is displayed in a tree structure in the **Repository location** list. Expand the folders until you find the object that you want to open. Select the object.
Note: You can set filter name and user information for the selected data sources by clicking the **Set Name Filter** and **Set User Information** buttons.
4. If you selected **From file** or **From repository** options, specify the number of the query result set in the **Result set** field.
5. Select **Always use this object when opening this document** check box to use the selected query for all other Analytics objects.
6. Click **OK** to close the **Data Source Object** dialog.
The source opens in the **Select Columns** dialog.
7. In the **Parameter selection** list, you can modify the available columns that refer to a particular chart axis. To modify a column, click the column that you want to modify and select the required one from the list.
8. Click **OK**.
The **Select Columns** dialog closes and the object is displayed on the **Results** tab of the editor area.

Results

You can export or print the opened Analytics object.

Using procedure commands with Analytics objects

You can open, export, and print Analytics objects by using the procedure commands.

About this task

For detailed information, see the topics:

- [DISPLAY command](#)
- [EXPORT command](#)
- [PRINT command](#)

Exporting Analytics objects

You can export opened Analytics objects as charts to a file system in the **Export Chart** wizard.

About this task

For more information, see [Exporting charts](#) topic.

Chapter 21. Customizing the application environment

There are many ways that you can customize your query and reporting environment.

Using the command line

The command line interface allows QMF for Workstation users to connect to a repository and run objects by submitting commands using their operating command line rather than the application interface.

About this task

The following steps describe how to connect to a secured repository and run objects using the operating system command line:

Procedure

1. Create a procedure.

For example:

```
run query ObjectTracking (form=ObjectTrackingForm
  export form to c:\dqtrack.frm
```

This example executes a query called *ObjectTracking* using the form called *ObjectTrackingForm* for form generation. The procedure exports the form to a file called *dqtrack.frm* in the *c:* directory.

2. Save the procedure in a secured repository with the name *proc12*, save the procedure in a QMF catalog with the name *DB2INST1.Procedure1*, and save as a local file with the name of *Procedure1.prc*.
3. There are several parameters that you can specify when you create your run command.

The following examples use some of the parameters and illustrate how to run a procedure object that has been saved in a repository, a QMF catalog and a local file:

- Type the following command in the operating system command line to connect to a secured repository when the login and password for the repository storage are saved in the repository connection named *Connection1*; the repository user ID *repuserid*; and the password *reppassword* and run a procedure named *proc12* that resides in the repository against the data source *DB2AIX* using the user ID *db2inst1* and password *db2inst1* to access the data source:

```
"C:\Program Files\IBM\QMF Analytics for Multiplatforms\
qmfddev.exe"
/RConnection:Connection1 /RUser:repuserid /RPassword:reppassword
/RObject:rsbi/.workspaces/Default/proc12 /IServer:DB2AIX /Run
/IUserID:db2inst1 /IPassword:db2inst1
```

- Type the following command in the operating system command line to connect to a secured repository when the login and password for the repository storage are not saved in the repository connection named *Connection1* and the repository storage user ID *repuserid* and password *reppassword* and the repository storage user ID *rdbuserid* and password *rdbpassword* and run a procedure named *DB2INST1.Procedure1* that has been saved in a QMF catalog against the data source *DB2AIX* using the user ID *db2inst1* and password *db2inst1* to access the data source:

```
"C:\Program Files\IBM\QMF Analytics for Multiplatforms\
qmfddev.exe" /RConnection:Connection1 /RUser:repuserid /RPassword:reppassword
/RDBUser:rdbuserid /RDBPassword:rdbpassword/IObject:DB2INST1.Procedure1
/IServer:DB2AIX /Run /IUserID:db2inst1/IPassword:db2inst1
```

- Type the following command in the operating system command line to connect to a repository with database-based security when the login and password for the repository storage are not saved in the repository connection named *Connection1* and the repository storage user ID *rdbuserid* and password *rdbpassword* and run a procedure that contains a global variable named

DB2INST1.Procedure1 that has been saved in a QMF catalog against the data source DB2AIX using the user ID db2inst1 and password db2inst1 to access the data source:

```
"C:\Program Files\IBM\QMF Analytics for Multiplatforms\qmfdev
.exe" /RConnection:Connection1 /RDBUser:rdbuserid
/RDBPassword:rdbpassword
/IObject:DB2INST1.Procedure1 /IServer:DB2AIX /Run /UserID:db2inst1
/IPassword:db2inst1
```

- Type the following command in the operating system command line to connect to a personal repository when the login and password for the repository storage are not required using the repository connection named Connection1 and run a procedure named Procedure1.prc that has been saved locally against the data source DB2AIX using the user ID db2inst1 and password db2inst1 to access the data source:

```
"C:\Program Files\IBM\QMF Analytics for Multiplatforms\
qmfdev.exe"
/RConnection:Connection1 c:\Procedure1.prc /IServer:DB2AIX
/Run /UserID:db2inst1 /IPassword:db2inst1
```

List of command line parameters

This list includes each of the operating system command line parameters and a description of the parameter's function.

In the descriptions of each parameter, an 'object' refers to a query, form, procedure, visual dashboard, or visual report.

| Parameter | Description |
|--|--|
| <i>ObjectName</i> | Use this parameter if the object that you will run resides in a local or network file. You must specify the full path name. |
| <i>/IObject:"ObjectOwner"."ObjectName"</i> | Use this parameter if the object that you will run resides in a QMF catalog. |
| <i>/RObject:pathtoobject</i> | Use this parameter if the object that you will run resides in a repository. You must specify the path to the object. For example: rsbi:/workspaces/default/query12 points to an object with the name query12 in a repository workspace called default. The path to any object in a repository is specified in the Key property found in the Properties window or view for the object. You open the Properties window by right-clicking an object in the Workspaces view. |
| <i>/RConnection:connectionname</i> | Name of the repository connection that will be used to connect to a repository. |
| <i>/IServer:servername</i> | The data source against which the object will be run. If this parameter is not specified, the last data source that was used will be accessed. |
| <i>/Run</i> | Start an opened or retrieved object. Note: If a form object does not have an associated data source, the <i>/Run</i> parameter is ignored. |
| <i>/IDisplay</i> | Display the object. If the <i>/Run</i> or <i>/IDisplay</i> parameters are not specified, then the <i>/IDisplay</i> parameter will be used. |

Table 36. Command line parameters (continued)

| Parameter | Description |
|--|--|
| <i>/RUser:ruserID</i> | The user ID that will be used to connect to any secured repository, except a repository with database-based security. |
| <i>/RPassword:rpassword</i> | The password that will be used with the user ID specified in the <i>/RUser</i> parameter. |
| <i>/RDBUser:rdbuserID</i> | The user ID that will be used to connect to the shared repository and authenticate user in the repository storage. |
| <i>/RDBPassword:rdbpassword</i> | The password that will be used with the user ID specified in the <i>/RDBUser</i> parameter. |
| <i>/IUserID:userID</i> | The user ID that will be used to access the data source against which the object will be run. |
| <i>/IPassword:password</i> | The password that will be used with the user ID specified in the <i>/IUserID</i> parameter. |
| <i>/Batch</i> | End the current session and close the application after running the object specified on the command line. Note: This parameter suppresses all warnings and error messages. |
| <i>"&variablename=variablevalue"</i> | Define or update global variable values for a procedure or query. Multiple variables can be defined. Note: The variable name and value sent to the procedure must be enclosed in double quotes (" ") and preceded by an ampersand (&). |
| <i>ICatalogUserID:cataloguserID</i> | The user ID that will be used to access an extended QMF catalog. |
| <i>ICatalogPassword:catalogpassword</i> | The user password associated with the <i>/ICatalogUserID</i> parameter. |

Using the command bar

You can run procedure commands directly from the QMF for Workstation interface using the command bar.

About this task

To run a procedure command from the command bar:

Procedure

1. Open the command bar by clicking the **Show Command Bar** toolbar button. If the command bar is closed it will be opened. If the command bar is already open and visible, clicking the icon will close it. You can also toggle the command bar open and closed by selecting **View > Command Bar**.
2. Enter the command that you want to run in the **Run command** field of the command bar. You can type the command in the field or select from a drop-down list of previously used commands. The following procedure commands are available:
 - BOTTOM
 - CONNECT

- CONVERT
 - DISPLAY
 - ERASE
 - EXPORT
 - FORWARD
 - IMPORT
 - PRINT
 - RESET GLOBAL
 - RUN
 - SAVE
 - SET GLOBAL
 - SHOW
3. Specify the name of a data source in the **Data source** field.

Specifying default print options

In QMF for Workstation, you must specify the page setup options that will be used when you print the contents of the available window.

About this task

If query results are in the window, then you will use the **Query Results Page Setup** dialog. If a form is in the available window, you will use the **Form Page Setup** dialog box.

To control the appearance of printed pages for query results and classic reports:

Procedure

1. Open the **Page Setup** dialog box by selecting **File > Page Setup**.
2. Specify the default printing options that will apply to your object.
3. Click **Set as Default** to save the setup values that you entered. The new values will become the default values.
4. Click **OK** to close the **Page Setup** dialog box. The options that you specified are used to control the appearance of the printed object.

Viewing resource limits

QMF resource limits control your data source access and resource consumption.

About this task

You can view the resource limits that are in effect for your user ID. You must be connected to the data source in order to view the resource limits that are in effect for your user ID when you access that data source.

To view resource limits for your user ID:

Procedure

1. Select **View > Resource Limits**. The **[Data Source]: Limits** window opens.
The information in the **Active Resource Limits** section shows whether the limits displayed are data source limits or query limits, and to what user and resource limit group they are applied.
2. Select the **Timeouts** page to see the warning and cancellation limits for idle queries, idle connections, and server response time that have been set for your user ID.

3. Select the **Limits** page to see the warning and cancellation limits for the maximum number of rows and bytes of data that can be retrieved using your current user ID and the maximum number of simultaneous connections QMF can establish to the database server for your user ID.
4. Select the **SQL Verbs** page to see the SQL verbs that your user ID is allowed to use when a particular schedule is in effect.
5. Select the **Options** page to see what server access options have been set for your user ID. The server access options include: database access using the QMF user interface and programming interface; database update confirmation; isolation levels for user queries; mandatory account information; and the ability to fetch all rows automatically, export data, edit tables, and only run saved queries.
6. Select the **Save Data** page to see if you are allowed to save data at the database.
7. Select the **Binding** page to see if you are allowed to bind or drop static packages for your queries.
8. Select the **Object Tracking** page to see if QMF object tracking is activated for your user ID.
9. Select the **LOB Options** page to see if you can access and retrieve data from tables containing large object data (LOB).
10. Select the **Cache** page to control caching options for each user.
11. Select the **Object list** page to define the default owner name to be used to filter data source objects.
12. Click **OK** to close the **Resource Limits for [user ID]** window.

About resource limits

Resource limits govern your access to a data source based on the user ID that you use to access the data source.

Your user ID can belong to multiple resource limits groups. In addition, your user ID can belong to multiple resource groups that are defined for multiple data sources. Each resource limits group that you belong to can have one or more schedules, each with a different set of resource limits. The schedule in effect specifies what associated set of resource limits are in effect for your user ID when you access a data source.

Most resource limits are checked before a query leaves the workstation. The query never reaches your target database for evaluation if one of these resource limits is violated. Other resource limits are evaluated after your target database has started to run the query. In this case, any database limiting rules take precedence over QMF resource limits.

The resource limits for your user ID are set by your QMF administrator. Only the person who administers QMF can change any of the resource limits values. All of the information displayed is read-only.

Specifying Timeouts resource limits

The **Timeouts** resource limits set the timeout warning and cancellation limits for idle queries, idle connections, and data source response time.

Idle Query Timeouts and **Server Response Timeouts** resource limits can be set for both data sources and for individual queries. The **Idle Connection Timeouts** resource limit can only be set for data sources and not for individual queries. The **Timeouts** resource limits are as follows:

Idle Query Timeouts

These fields specify the amount of time that a query can remain idle. The **Warning limit** field displays the number of seconds that a query can remain idle before the user is informed and asked if the query should be canceled. The **Cancel limit** field displays the number of seconds that a query can remain idle before the query is automatically canceled.

Server Response Timeouts

These fields specify the amount of time that can be spent waiting for a response from the database. The **Warning limit** field specifies the number of seconds that can elapse before the user is informed and asked if the database request should be canceled. The **Cancel limit** field specifies the number of seconds that can elapse before the database request is automatically canceled.

Note: A lower timeout limit prevents long running, runaway queries. A higher timeout limit allows database requests to complete when the database server is slow due to resource contention or other reasons.

Idle Connection Timeouts

This field specifies the amount of time that an idle connection to a database data source is retained. The **Cancel limit** field specifies the number of seconds that an idle connection is retained before it is automatically closed.

Note: A lower timeout limit minimizes the resources consumed at the database by idle connections. A higher timeout limit minimizes the overhead of establishing connections.

Specifying Limits resource limits

The **Limits** resource limits set the warning and cancellation limits for the maximum number of rows and bytes of data that can be retrieved from the database data source; and the maximum number of simultaneous connections that can be established to the data source.

Limits resource limits can be set for data sources. The **Limits** resource limits are as follows:

Maximum Rows to Fetch

These fields specify the limits for the number of rows of data that can be retrieved from a database data source when running a query. The **Warning limit** field specifies the number of rows that can be retrieved from the data source before you are notified that a row limit has been reached. The user is prompted to select whether more data should be fetched or should the query be canceled. The **Cancel limit** field displays the number of rows that can be retrieved from the data source before the row limit is reached. The query is then automatically canceled.

Maximum Bytes to Fetch

These fields specify the limits for the number of bytes of data that can be retrieved from a database data source when running a query. The **Warning limit** specifies the number of bytes that can be retrieved from the data source before the user is notified that the byte limit has been reached. The user is then prompted to specify whether more data should be fetched or should the query be canceled. The **Cancel limit** displays the number of bytes that can be retrieved from the data source before the byte limit is reached. The query is then automatically canceled.

Maximum Connections

This field specifies the limit for the number of simultaneous connections that can be established to a database data source. The **Cancel limit** specifies the number of simultaneous connections that can be established to a data source before the query requesting the next connection is automatically canceled. Setting the limit to zero (0) indicates that there is no limit to the number of connections.

Generally, connections to database data sources are reused. If you run one query at a data source, then run another query at the same data source, only one connection is required. However, if you run those two queries simultaneously, then two connections are required. If another connection to a data source is required and this limit is reached, an error is returned and the operation is not performed.

Note: To enable table editing, you must specify that two or more simultaneous connections are allowed.

Specifying SQL Verbs resource limits

The **SQL Verbs** resource limits specify the SQL verbs that you are allowed to use when the current schedule is in effect.

If a user tries to run a query that contains a *disallowed verb*, the query is canceled without sending the SQL to the data source. **SQL Verbs** resource limits can be set for data sources.

The use of the following SQL verbs can be controlled:

- ACQUIRE
- ALTER
- CALL

- COMMENT
- CREATE
- DELETE
- DROP
- EXPLAIN
- GRANT
- INSERT
- LABEL
- LOCK
- REFRESH
- RENAME
- REVOKE
- SET
- SELECT
- UPDATE

Specifying Options resource limits

The **Options** resource limits specify the levels of access you have to a data source and the data source objects.

Options resource limits can be set for data sources. The **Options** resource limits are as follows:

Allow access to data source from user interface

This field specifies whether you have permission to access a data source from the QMF for Workstation user interface when the current resource limits group schedule is in effect.

Allow access to data source from programming interface

This field specifies whether you have permission to access a data source from the QMF for Workstation application programming interface (API) when the current resource limits group schedule is in effect.

Fetch all rows automatically

This field specifies how the data source will send query results data to the application. By default, data is requested from the data source only as required to display query results. For example, if 20 rows fill up the query results window, only 20 rows are requested from the data source. When you scroll down to make the 21st row visible, more data is requested. If you run a query and then wait before scrolling down, the query remains active for that wait time, consuming resources at the data source. If this parameter is enabled, data is requested repeatedly until all data is received, independent of your scrolling requests.

Confirm database updates

This field specifies whether or not you must confirm all data source changes resulting from the queries that you run or the actions that you perform when editing tables. If this check box is selected, you will be prompted for confirmation before any data source changes can occur. If this check box is cleared, data source changes can occur without confirmation.

You can override this resource limit setting using the **Confirm** parameter in a **Procedure** command. The following table shows the correlation of the parameters, depending on the specified values.

| Confirm database updates value | Procedure parameter value | The confirmation dialog behavior |
|--------------------------------|---------------------------|----------------------------------|
| Selected | confirm=yes | appears |
| Selected | confirm=no | does not appear |

| Confirm database updates value | Procedure parameter value | The confirmation dialog behavior |
|--------------------------------|---------------------------|----------------------------------|
| Selected | not specified | appears |
| Cleared | confirm=yes | appears |
| Cleared | confirm=no | does not appear |
| Cleared | not specified | does not appear |

Enable exporting of data

This field specifies whether a user can export query result data to a file using the **Export** command from the **File** menu or the **EXPORT DATA** procedure command.

Enable table editing

This field specifies whether or not you can edit a table using the table editor feature .

Allow running saved queries only

This field specifies whether or not you can create and save new queries at the database data source. If checked, you can only run queries that are saved at the database data source. If clear, you can run queries, as well as create and save new queries at the database data source.

Isolation level for queries

The **Isolation level for queries** parameter only applies to databases that support transactions.

This field specifies the level of transaction isolation to be applied to a user's queries. There are varying levels of isolation, each with a differing degree of impact on transaction processing in the database.

Consult with your database administrator or the database server documentation to determine if the database server being used is configured to support transaction processing.

Valid values include:

- **Default** specifies that the isolation level for a query will be the default specified for all applications.
- **Repeatable Read** specifies that the isolation level for a query will be Repeatable Read (RR). The RR isolation level provides the most protection from other applications.
- **All (RS)** specifies that the isolation level for a query will be Read Stability (RS).
- **Cursor Stability (CS)** specifies that the isolation level for your query will be Cursor Stability (CS). The CS isolation level provides the maximum concurrency with data integrity.
- **Change (UR)** specifies that the isolation level for your query will be Uncommitted Read (RS) which allows an application to read uncommitted data.
- **No Commit (NC)** specifies that the isolation level for your query will be No Commit (NC). An application running with an isolation level of NC can not make permanent database changes.

Account

This field specifies the default account information that will be sent to a data source when you connect to that data source.

Account can be overridden

This field specifies whether you are allowed to override the default accounting information specified in the **Account** field when connecting to a data source. If selected, you can enter a new accounting string using the **Set User Information** window. If clear, you cannot enter new accounting information and you are prevented in some cases from accessing data source objects that are not owned by your resource limits group.

Require account information

This field specifies whether you must specify a valid accounting string in the **Set User Information** window before connecting to a database when this schedule is in effect.

Allow printing

This check box must be selected in order for you to print objects that are stored on the data source.

Specifying Save Data resource limits

The **Save Data** resource limits specify whether you as a member of the current resource limits group and using the current schedule can save data at the database data source.

Save Data resource limits can be set for data sources. The **Save Data** resource limits are as follows:

Allow save data command

This field specifies whether you can save data at the database data source. Saving data can be extremely resource-intensive. As a result it can have a significant impact on your database and network.

Default table space

This field specifies the default table space name that will be used for tables created with the save data command. This field is only available if the **Allow save data command** has been selected. The syntax of the table space name that you specify must conform to the database rules for table space names. Any value that you specify is used as part of the CREATE TABLE SQL statement that runs when you save data to a new table. Entering an asterisk (*) specifies that the default table space will be taken from a user's profile in the RDBI . PROFILE_TABLE or Q . PROFILES that are specified by the administrator when your profile was created.

Default table space can be overridden

This field specifies whether you can override the default table space. If selected, you can specify any table space name subject to database security authorizations. If this check box is not selected, you must use the table space name specified in the **Default table space** field.

Allow to replace any existing data

This field specifies whether a user can override the existing data when the objects are exported or imported to the database. If this option is selected, a user can replace the existing data with the new one in the database. If this check box is clear, a user can only append new data to the existing one.

Specifying Binding resource limits

The **Binding** resource limits specify whether you as a member of this resource limits group and using this schedule can bind or drop static packages for your queries to or from the data source.

Binding resource limits can only be set for data sources. The **Binding** resource limits are as follows:

Allow binding of packages

This field specifies whether or not you can bind static packages for your queries on the database data source.

Allow dropping of packages

This field specifies whether or not you can drop static packages from the database data source.

Default collection ID

This field specifies the default collection ID for the static packages that you bind.

Default collection ID can be overridden

This field specifies whether you must use the default collection ID. If this check box is selected, you can specify any collection ID subject to database security authorizations. If this check box is clear, you must use the default collection ID that is specified in the **Default collection ID** field.

Default isolation level

This field specifies the default isolation level for the static packages that you bind. You can select one of the isolation levels:

- **Repeatable Read** specifies that the default isolation level for the static packages will be Repeatable Read (RR). The RR isolation level isolates the SQL statements in the static package from the actions of concurrent users for rows the requester reads and changes, including phantom rows. RR isolation provides the most protection.
- **All (RS)** specifies that the default isolation level for the static packages will be Read Stability (RS). The RS isolation level isolates the SQL statements in the static package from the actions of concurrent users for rows the requester reads and changes

- **Cursor Stability (CS)** specifies that the default isolation level for the static packages will be Cursor Stability (CS). The CS isolation level isolates the SQL statements in the static package and the current row to which the database cursor is positioned from the actions of concurrent users for changes the requester makes.
- **Change (UR)** specifies that the default isolation level for the static packages will be Uncommitted Read (UR). The UR isolation level isolates the SQL statements in the static package from the actions of concurrent users for changes the requester makes.
- **No Commit (NC)** specifies that the default isolation level for the static packages will be No Commit (NC). The NC isolation level does not isolate the SQL statements in the static package from the actions of concurrent users for changes the requester makes.

Default isolation level can be overridden

This field specifies whether you can override the isolation level specified in the **Default isolation level** field. If this check box is selected, you can specify any isolation level when binding packages. If this check is not selected, you must use the default isolation level specified in the **Default Isolation Level** field to bind static packages.

Specifying Object Tracking resource limits

The **Object Tracking** resource limits specify whether object tracking is activated for your user ID.

Object tracking must be enabled in order to collect the tracking data that is used to populate object tracking reports. Object tracking reports can help you locate unused objects, locate frequently accessed data sources such as tables or columns and spot potential problem areas. **Object Tracking** resource limits can be set for data sources.

The **Object Tracking** resource limits are as follows:

Enable summary object tracking

This field specifies whether summary object tracking is activated for your resource limits group. Summary tracking records the number of times an object is run, and the most recent times it was run and modified.

Enable detailed object tracking

This field specifies whether detailed object tracking is activated for your resource limits group. Detailed tracking records each time the object is run, who ran it, and the results. A large amount of data can be collected if this option is selected.

Note: This option must be enabled to use *ad hoc* object tracking or SQL text tracking.

Enable ad-hoc object tracking

This field specifies whether a record of each ad-hoc query is kept in the summary or detail object tracking tables. An ad-hoc query is a query that has not been named or saved at the database data source.

Note: The **Enable detailed object tracking** option must be enabled in order to enable this option.

Enable SQL text tracking

This field specifies whether a record of the SQL text of each query is kept in the detail tracking table.

Note: The **Enable detailed object tracking** option must be enabled in order to enable this option.

Last Used timestamp option

This field specifies when and by which commands data objects are updated.

Note: In cases when the object is saved to the QMF Catalog, the data is kept in the column **LAST_USED** of the table **Q.OBJECT_DIRECTORY**. If the object is saved to the repository, then the data is kept in the column **LAST_USED** of the **RSBI.OBJECTS** table.

The following options are available:

- **Update when object is accessed (legacy mode)**

Data is updated when you run any of the following commands against the object:

- **CONVERT**
- **DISPLAY**
- **EXPORT**
- **IMPORT**
- **PRINT**
- **RUN**
- **SAVE**

- **Update when object is run, saved or imported**

Data is updated after performing **RUN**, **SAVE** or **IMPORT** commands.

- **Update only when object is run**

Data is updated only after performing **RUN** command.

Last Used timestamp options can be overridden

Indicates whether the DSQEC_LAST_RUN variable value is ignored.

This check box is cleared by default.

Select this check box to change the variable value and override the resource limit.

Specifying LOB Options resource limits

The **LOB Options** resource limits control a users ability to access, retrieve, and save data to and from tables containing large objects (LOBs).

A Large Object (LOB) is a Db2 for Linux, UNIX, and Windows data type that houses non-traditional data such as text, multimedia, image, video, photograph, sound, or any very large data file inside a database table. Retrieving or saving LOB data can consume a substantial amount of resources. Even just accessing a table that contains LOB data can be resource consumptive. Using the **LOB Options** resource limits, you can control access to tables on the database data source that include LOB data. **LOB Options** resource limits can be applied to both data sources and individual queries.

The **LOB Options** resource limits are as follows:

LOB Retrieval Option

This field specifies whether a user can retrieve large object (LOB) data when this schedule is in effect. You can select one of the following:

- **Disable LOB columns** specifies that users can not retrieve LOB data and can not query or access any table that contains LOB data.
- **Disable LOB data retrieval** specifies that users can not retrieve LOB data but can query tables that contain LOB data. Query result data will be returned for all columns in the table except those that contain LOB data.
- **Retrieve LOB data on demand** specifies that users can query tables that contain LOB data and result data will be returned for all columns including those columns that contain LOB objects. However, the results data for the LOB columns will not actually be displayed in the column. A pointer is displayed. When the user selects the pointer, the LOB data will be displayed in the query results.

You can also pull the data from the database by double-clicking on the pointer.

- **Retrieve LOB data automatically** specifies that users can query tables that contain LOB data and the result data will be returned for all columns, including those columns that contain LOB objects.

This option pulls ALL of the LOB data for ALL of the LOB columns from the database to the local computer.

This option can potentially consume a large amount of resources and time. The actual LOB data is not displayed in grids and reports. Instead, pointers to the LOB data are displayed.

Enable saving LOB data

This field specifies whether users can save LOB data on the database data source.

Maximum LOB column size

This field specifies the maximum size of a LOB column, in kilobytes, up to 2G (the maximum LOB size). The default is 0, no maximum. If a user queries a table with LOB data that is larger than the maximum, the LOB data will not be returned for display.

LOB options can be overridden

This field specifies whether a user can override the default LOB Options resource limits that have been specified for their group.

Specifying Cache resource limits

The **Cache** resource limits control caching for query results.

With this resource limit enabled, when a query is run, the results will be cached. If the query is run again, and nothing has changed, the results will be taken from the cached results rather than run again at the data source. **Cache** resource limits can be applied to both data sources and individual queries.

The **Cache** resource limits are as follows:

Cache is enabled

This field specifies whether the caching of result set data is enabled. Caching is disabled by default. If caching is enabled, result data from a query is cached (stored) and is available for use by other users that have the same privileges to access this data on the data source. Caching result set data can significantly decrease resource consumption and improve performance. The cached result set data is not automatically refreshed.

Auto data fetching enabled

This field specifies how the cached rows of data will be retrieved. This resource limit becomes available if **Cache is enabled** is selected. When selected, this option specifies that if the query result set is taken from the cache, the entire result set will be presented. When this option is not selected, 100 rows (the default) of the query result set is presented.

Data expiration interval

This field specifies the time period the query result set data will remain in cache. You can set the time period by specifying the number of days, hours, and minutes. After this period, the cached query result set data will be updated.

Cache options can be overridden

This field specifies whether the cache options can be overridden.

Specifying Object List resource limits

The **Object List** resource limits specifies default and allowed owner names for data sources.

With **Object List** resource limits, you can define the default owner name that will be used to filter data source objects. You can also specify a specific list of owner names that can be used to filter data source objects aside from the default name.

The **Object List** resource limits are as follows:

Default Owner Name

This field specifies the default owner name that will be used to filter the data source object list.

Default Owner Name can be overridden

This check box allows the user to override the current default owner name.

Allowed Owner Names

This field specifies the list of additional owner names that can be used to filter the data source object list. Owner names should be separated by a comma.

Allow <USERNAME>

This check box allows the current user's owner name to be used to filter the data source object list.

Allow <SQLID>

This check box allows the current user's SQLID to be used to filter the data source object list.

Appendix A. Accessibility

Software accessibility features help those with physical disabilities, such as restricted mobility or limited vision, use their computers.

Note: The QMF for Workstation application provides the best accessibility functionality. Users with physical disabilities who must make use of the QMF accessibility features should install and run QMF for Workstation, as opposed to QMF for WebSphere.

Standard keyboard equivalents in QMF for Workstation

Keyboard equivalents use keyboard keys to perform mouse actions instead of using a mouse.

The keyboard is the most frequently used alternative for performing mouse functions. QMF supports the standard keyboard equivalents that include:

- Shortcut, or accelerator keys, to perform the most frequently used functions in pull-down menus without going to the menu. For example:
 - **Ctrl+S** to save
 - **Ctrl+P** to print
 - **Ctrl+R** to run a query

Shortcuts display in the pull-down menu next to its function.

- Mnemonics, or access keys, are available to perform each function on a menu or window. A mnemonic for a function is the underlined character in the function name. For example:
 - Press F to open the **File** menu
 - Press O to open the **Open** window

Use **ALT** to activate the mnemonic and move the keyboard focus.

For more information, refer to the documentation for your operating system for a complete list of standard keyboard equivalents.

Standard keyboard navigation

Keyboard navigation refers to using keys instead of a mouse to move from item to item on your screen.

The movement is usually in the order specified by the operating system or your application. QMF applications follow the standards with regard to the typical keys used for keyboard navigation such as using **TAB** and **SHIFT+TAB** to move between controls, and arrow keys to move up, down, and sideways between items.

Operating system accessibility

Most operating systems have a set of accessibility options that enable individuals with disabilities to customize system-wide settings.

QMF applications:

- Inherit their settings from the operating system
- Do not interfere with keyboard accessibility features built into the operating system

Assistive technology products

QMF applications support assistive technology products, such as screen readers and voice synthesizers.

Each QMF application interface requires special navigation when using a screen reader with query results. See [Navigation in QMF for Workstation](#) for more information.

Navigation in QMF for Workstation

The following table shows how to accomplish various tasks using keystrokes.

| To: | Process: |
|---|---|
| Activate mnemonics in menus | Press the Alt key. Mnemonics in each menu are activated. |
| Obtain online help | Press F1 to open a specific help task or Alt+H to open the Help menu. Use the arrow keys to move through the menu selections. Press Enter to open a menu command. |
| Display keyboard shortcuts | Press Ctrl+Shift+L . |
| Display the Draw Query wizard | Press Ctrl+D . |
| Display the switch list of the editors of the opened QMF objects and switch between the editors | Press Ctrl+F6 to move forward in the switch list and Ctrl+Shift+F6 to move in the opposite direction. |
| Left to right switch between the tabs of the opened QMF objects | Press Ctrl+Page Up |
| Right to left switch between the tabs of the opened QMF objects | Press Ctrl+Page Down |
| Move focus through each element | Press TAB to move forward or SHIFT+TAB to move in the opposite direction. |
| Emulate clicks on a link | Use the TAB key to jump from link to link, then press the Enter key to open the link. |
| Emulate clicks on a button | Use the TAB key to jump from button to button, then press Space key to activate the button. |
| Within a window, activate the default action | Press the Enter key. |
| Within a window, cancel the action | Press the ESC key. |
| Left to right switch between views | Press Ctrl+F7 |
| Right to left switch between views | Press Ctrl+Shift+F7 |
| Left to right switch between perspectives | Press Ctrl+F8 |
| Right to left switch between perspectives | Press Ctrl+Shift+F8 |

| To: | Process: |
|--|--|
| Left to right switch between tabs in the active editor | Press Alt+Page Up |
| Right to left switch between tabs in the active editor | Press Alt+Page Down |
| Activate command bar | Press Ctrl+Shift+C |
| <p>Use screen reader for query results</p> <p>or</p> <p>Use browser to display query results in high contrast mode for the visually impaired</p> | <ol style="list-style-type: none"> 1. Run the query. 2. Press Alt+R to open the Results menu 3. Select Export to open the Export Query Results wizard. 4. Use the TAB key to move through controls. Use arrow keys to move through selection items in the control. Select HTML from the Export Type list. Specify where to save the file in the File name field. 5. Press Finish. 6. Open the *.htm file in a browser. <p>or</p> <ol style="list-style-type: none"> 1. Run the query. 2. Press Alt+R to open the Results menu. 3. Select Export to open the Export Query Results wizard.. 4. Use the TAB key to move through controls. Use arrow keys to move through selection items in the control. Select CSV from the Export Type list. Specify where to save the file in the File name field. 5. Press Finish. 6. Open the *.csv file in supportive software such as Microsoft Excel. |
| Add object or condition | Use TAB to switch between different sections of the Query Builder editor, and then press Spacebar to expand the sections. |
| Select multiple objects | <ol style="list-style-type: none"> 1. Tab to the list box. 2. Press Shift+Arrow to select additional, consecutive rows. 3. Press Shift+Enter to add the objects. |
| Edit rows in list boxes (only for rows that can be edited) | <ol style="list-style-type: none"> 1. TAB to the list box, such as the Global Variables window. 2. Press the spacebar or use the up and down arrow keys to select the row. 3. Press F2 to activate edit mode. 4. Use TAB to move between columns and rows. 5. Press Enter to accept edits. 6. Press TAB to exit the list box and move to the next control in the dialog. |

| To: | Process: |
|-------------------------------|--|
| Edit query results | <ol style="list-style-type: none"> 1. Press CTL+F7 or CTL+SHIFT+F7 until either the Workspaces or Repository Explorer view is highlighted. 2. Use the down or up arrow keys to select the folder that contains a table. 3. Use the right arrow key to expand folders and the left arrow key to collapse folders. 4. Use the arrow keys to expose and select the table that you want to edit. 5. Press the Menu key (the key to the left of the right-side of the CTRL key). 6. Press the mnemonic key for Open With (the h key). 7. Use the down arrow key to highlight Table Editor. 8. Press the ENTER key. The table opens in Table Editor. |
| Change font for query results | <ol style="list-style-type: none"> 1. Run the query. 2. Press Alt+A to select all. 3. Press Alt+R to open the Results menu. 4. Press F to open the Font window. |

Navigation in QMF for WebSphere

The following table shows how to accomplish various tasks using keystrokes.

| To: | Process: |
|--|--|
| Move focus through each element | Press TAB to move forward or SHIFT+TAB to move in the opposite direction. |
| Emulate clicks on a link | Use the TAB key to jump from link to link, then press the Enter key to open the link. |
| Emulate clicks on a button | Use the TAB key to jump from button to button, then press Space key to activate the button. |
| Within a dialog, activate the default action | Press the Enter key. |
| Within a dialog, cancel the action | Press the ESC key. |
| Move focus to the command line | Press ALT+C . |
| Move focus to the main menu | Press the left ALT key. Navigate through the menu commands using the arrow keys. |
| Open context menus | Press the pop-up menu key if you have an extended keyboard. Press the right Ctrl key. Navigate through the menu commands using the arrow keys. |
| Activate top-level menu | Press Shift+Ctrl+1 (Shift+Ctrl+2, Shift+Ctrl+3, and so on) |
| Switch between open editors | Press Shift+Ctrl+F6 |
| Switch between open views | Press Shift+Ctrl+F7 |

Appendix B. Troubleshooting

Use these topics to diagnose and correct problems that you experience with QMF.

Problem displaying images in QMF for WebSphere using Internet Explorer 8

Symptoms

Visual report and visual dashboard images do not display properly in QMF for WebSphere when using Internet Explorer version 8.

Environment

QMF for WebSphere

Resolving the problem

Delete files from the **Temporary Internet files** folder in the Internet Explorer browser:

1. Select **Tools > Delete Browsing History** from the menu bar. The **Delete Browsing History** dialog opens.
2. Check the **Temporary Internet files** check box.
3. Click **Delete**. The **Delete Browsing History** dialog closes and the files are deleted.
4. Reopen the visual report or visual dashboard.

Creating a response file

To run a silent installation of QMF for Workstation, you must create a response file that specifies all installation options. This file must specify all the features that you want to install and installation paths.

About this task

To create a response file:

Procedure

1. In the command line, run the installation command.
 - See the examples for a Windows installation to the current directory:

```
setupwin32.exe -r installer.properties
```

or

```
setupwin64.exe -r installer.properties
```

where: *-r* specifies that the installation software runs in record mode.
 - See the example for a Windows installation to the custom directory:

```
setupwin32.exe -r C:\drive_path\response_file.properties
```

or

```
setupwin64.exe -r C:\drive_path\response_file.properties
```

where:
 - *-r* specifies that the installation software is run in the record mode.

- C:\drive_path\response_file.properties specifies the path to the response file that you want to create.
2. As the installation proceeds, you select the features that each user that runs this executable installs on their machine.
 3. Upon completion of the installation, the application is installed in the directory that you specified during the installation.

Results

The response file is created and saved in the current directory or the directory that you specified in the path.

What to do next

Distribute the files to the user machines to that you want to install the software and run the silent installation.

QMF for Workstation Win32 API reverts to previous version of QMF for Windows

Symptoms

If an earlier version of QMF for Windows that is installed alongside QMF for Workstation Version 12 is opened, API instances automatically revert to QMF for Windows, even after Version 12 has been reopened.

Causes

When QMF for Workstation Version 12 is installed, the application version number is updated in the system registry. When a preexisting version of QMF for Windows is opened, it automatically reverts the application's version number in the system registry. When Version 12 is opened, this automatic rename does not take place.

Resolving the problem

Any time that the QMF for Windows application is opened, you must manually reset the version number in the system registry if you want API instances to point to QMF for Workstation Version 12.

User response: To reset the application version number in the system registry:

1. Go to the following location in your workstation directory:

```
"C:\Program Files\IBM\QMF Analytics for Multiplatforms\API
```

2. Double-click the file named qmfwin.exe

Note: Win32 API uses active repository connections. If there is not an active repository connection, the system brings up a dialog box that lists all available connections where you must select the connection you want to use.

Appendix C. Messages

This topic lists and describes the messages and return codes issued by the program components of QMF for Workstation and QMF for WebSphere.

QMF for Workstation and QMF for WebSphere messages are numbered as follows:

*CPT*prefix*MESS*number

CPTprefix

The abbreviation of the component of the application that threw the error. This can be an indication of the first place to look in order to resolve the error.

MESSnumber

The specific number of the error message.

For example: ABC0001

All of the error messages of each component are listed in alphabetical order. To look up more information about a specific error message and how to resolve it, search for the component prefix and number of the message in this publication.

Note: Since QMF supports Db2 databases, Db2 error messages are included in the application. Db2 error messages have a component prefix of DB2. (for example, DB20001) If you encounter a message with this component prefix, refer to the Db2 online documentation at www.ibm.com. If you are using the DB2 Type 4 UDB driver or the DB2 Universal JDBC Driver, to receive extended SQL exception error messages, check the **retrieveMessagesFromServerOnGetMessage** option in the **Advanced JDBC Settings** window of the **Edit Repository Connection** wizard.

API1001 {0}

Explanation:

This is a container for other messages.

User response:

No action is required.

API1002 **The API has not been initialized.**

Explanation:

The API instance has not been initialized. This could mean that the QMFwin API was not initialized, or a repository connection was not established.

User response:

Check the QMFwin API logs to determine the error that caused the API instance to fail, or establish a repository connection.

API1003 **No data found.**

Explanation:

There was an attempt to obtain a row that does not exist in a result set.

User response:

No action required.

API1006 **The specified source {0} is not valid.**

Explanation:

An invalid value was specified for the SourceType parameter of the initializeProc() or initializeQuery() function.

User response:

Specify a valid value for the SourceType parameter of any initializeProc() or initializeQuery() function.

API1007 **The specified source name is not valid.**

Explanation:

An invalid value was specified for the Source, CollectionName, PackageName, or ConsistencyToken parameter of an initializeProc(), initializeQuery(), or initializeStaticQuery() function.

User response:

Specify a valid value for the Source, CollectionName, PackageName, or ConsistencyToken parameter of any initializeProc(), initializeQuery(), or initializeStaticQuery() function.

API1008 **The specified database server {0} is not valid.**

Explanation:

An unknown data source alias was specified in a getServerAttributes(), getServerType() or initializeServer() function.

User response:

Specify a valid data source alias for any `getServerAttributes()`, `getServerType()` or `initializeServer()` function.

API1009 **No database server has been specified.**

Explanation:

The `initializeServer()` function was not invoked before other functions were invoked that require an initialized data source.

User response:

Ensure that the `initializeServer()` function is invoked before other data source dependent functions are called.

API1010 **The specified row limit {0} is not valid.**

Explanation:

An invalid value was specified for the `RowLimit` parameter of an `open()` function.

User response:

Specify a positive value for the `RowLimit` parameter of any `open()` function.

API1011 **The specified query ID {0} is not valid.**

Explanation:

An invalid value was specified for the `QueryID` parameter of an API function.

User response:

Specify a valid value for the `QueryID` parameter of all relevant functions.

API1012 **The SQL verb {0} cannot be used with open().**

Explanation:

An `open()` function was called for a query that does not contain the `SELECT`, `VALUES`, `WITH`, or `CALL` SQL verbs.

User response:

Ensure that any `open()` function calls a query that contains one of the SQL verbs that was mentioned above.

API1013 **The SQL verb {0} cannot be used with execute().**

Explanation:

An `execute()` function was called for a query that contains the `SELECT`, `VALUES`, `WITH`, or `CALL` SQL verbs.

User response:

Ensure that any `execute()` function does not call a query that contains one of the SQL verbs that was mentioned above.

API1015 **A new procedure could not be created. There may be too many documents already open, or there may be insufficient memory.**

Explanation:

An unexpected error occurred while creating a procedure using the `initializeProc()` function.

User response:

Check the QMFwin API logs to find the root error.

API1016 **A new query could not be created. There may be too many documents already open, or there may be insufficient memory.**

Explanation:

An unexpected error occurred while creating a query using the `initializeQuery()` function.

User response:

Check the QMFwin API logs to find the root error.

API1024 **The specified column delimiter ({0}) is not valid.**

Explanation:

An invalid value was specified for the `ColumnDelimiter` parameter of an `export()` function.

User response:

Specify a valid value for the `ColumnDelimiter` parameter of any `export()` function.

API1026 **The specified file name is not valid.**

Explanation:

An invalid or empty value was specified for the `FileName` parameter of an `export()` function.

User response:

Specify a valid value for the `FileName` parameter of any `export()` function.

API1027 **The specified format ({0}) is not valid.**

Explanation:

An invalid value was specified for the `Format` parameter in an `export()` function.

User response:

Specify a valid value for the `Format` parameter of any `export()` function.

API1028 **The specified form name is not valid.**

Explanation:

An invalid or empty value was specified for the `Form` parameter of a report related function.

User response:

Specify a valid value for the Form parameter of any report related functions.

API1029 **The specified form source ({0}) is not valid.**

Explanation:

An invalid value was specified for the SourceType parameter of a report related function.

User response:

Specify a valid value for the SourceType parameter of any report related function.

API1030 **The global variable name is invalid.**

Explanation:

An invalid value was specified for the GlobalVariableName parameter of a global variables related function.

User response:

Specify a valid value for the GlobalVariableName parameter of any global variables related function.

API1031 **The specified host variable is not valid.**

Explanation:

An invalid value was specified for the Value parameter of a setHostVariable() function.

User response:

Specify a valid value for the Value parameter of any setHostVariable() function.

API1033 **The specified owner and name ({0}) are not valid.**

Explanation:

An invalid value was specified for the OwnerAndName parameter of an API function.

User response:

Specify a valid value for the OwnerAndName parameter of any relevant API function.

API1034 **The specified page length ({0}) is not valid.**

Explanation:

An invalid value was specified for the PageLength parameter of an exportReport() or exportVisualReport() function.

User response:

Specify a valid value for the PageLength parameter of any exportReport() or exportVisualReport() functions.

API1035 **The specified page width ({0}) is not valid.**

Explanation:

An invalid value was specified for the PageWidth parameter of an exportReport() or exportVisualReport() function.

User response:

Specify a valid value for the PageWidth parameter of any exportReport() or exportVisualReport() function.

API1037 **The specified procedure ID ({0}) is not valid.**

Explanation:

An invalid value was specified for the ProcID parameter of a procedure related function.

User response:

Specify a valid value for the ProcID parameter of any procedure related functions.

API1038 **The specified resource ({0}) is not valid.**

Explanation:

An invalid value was specified for the Resource parameter of a getResourceLimit() function.

User response:

Specify a valid value for the Resource parameter of any getResourceLimit() function.

API1039 **The specified range of rows and columns is not valid.**

Explanation:

Invalid values were specified for the FirstRow, FirstCol, LastRow, and LastCol parameters of an exporting result set function.

User response:

Specify valid values for the FirstRow, FirstCol, LastRow, and LastCol parameters of any exporting result set function.

API1040 **The specified string delimiter ({0}) is not valid.**

Explanation:

An invalid value was specified for the StringDelimiter parameter of an export() function.

User response:

Specify a valid value for the StringDelimiter parameter of any export() function.

API1041 **The specified table name is not valid.**

Explanation:

An invalid value was specified for the TableName parameter of an export() function.

User response:

Specify a valid value for the TableName parameter of any export() function.

API1042 **The specified target server is invalid.****Explanation:**

An invalid value was specified for the Server parameter of a saveData() or copyQMFOBJECT() function.

User response:

Specify a valid value for the Server parameter of any saveData() or copyQMFOBJECT() function.

API1044 **The specified time ({0}) is not valid.****Explanation:**

An invalid value was specified for the Time parameter of a getQMFOBJECTINFO() function.

User response:

Specify a valid value for the Time parameter of any getQMFOBJECTINFO() function.

API1045 **The specified type ({0}) is not valid.****Explanation:**

An invalid value was specified for the Type parameter of a getQMFOBJECTINFO() function.

User response:

Specify a valid value for the Type parameter of any getQMFOBJECTINFO() function.

API1046 **The specified variable ({0}) is not valid.****Explanation:**

An invalid value was specified for the VariableName parameter of a setVariable() function.

User response:

Specify a valid value for the VariableName parameter of any getVariable() function.

API1053 **The SQL verb ({0}) cannot be used with ExecuteStoredProcedure().****Explanation:**

A query that does not contain the CALL SQL verb was specified for an executeStoredProcedureEx() function.

User response:

Specify a CALL query for any executeStoredProcedureEx() function.

API1054 **Can't prepare a static section.****Explanation:**

A static query was specified for a prepare() function; only dynamic queries can be specified.

User response:

Specify a dynamic query for any prepare() function.

API10004 **The values specified for ForceDialog and SuppressDialog are mutually exclusive.****Explanation:**

The initializeServer() function was called and the values that were specified for ForceDialog and SuppressDialog parameters were mutually exclusive.

User response:

Check the code that invokes the initializeServer() function.

API10005 **The specified value is not valid.****Explanation:**

The value that was specified for the parameter of a function is incorrect.

User response:

Check to ensure that all of the specified parameter values for all of the functions that were called in the API instance are valid.

BIC00000 **Internal Error****Explanation:**

There was an internal error.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

BIC00001 **Illegal Argument****Explanation:**

The specified argument is not valid.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

BIC00002 **Null Pointer****Explanation:**

A null pointer exception occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

BIC00004 **{0}****Explanation:**

The specified exception occurred.

User response:

Correct the specified exception and try again.

BIC00005 **Unknown exception.****Explanation:**

An error occurred while retrieving a resource string.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

BIC00006 This feature is not supported by QMF

Explanation:

The specified feature is not supported by QMF

User response:

No action is required.

BIC00007 Object is not supported by QMF

Explanation:

The specified object is not supported by QMF

User response:

No action is required.

BIC00009 Fetching of children is cancelled by user.

Explanation:

The operation was cancelled by the user.

User response:

Retry to fetch the specified objects.

CLC00001 Cannot convert {0} to number.

Explanation:

The specified element can not be converted into a numeral.

User response:

Specify valid element to convert into a numeral.

CLC00002 Unable to evaluate linReg: data series have different numbers of elements.

Explanation:

The data series specified for the x and y variables contain different numbers of data points.

User response:

Specify data series that contain the same number of data points.

CLC00003 Unable to evaluate linReg: not enough data.

Explanation:

The data series specified do not contain enough rows to calculate the linear regression.

User response:

Specify a query result set that contains more than one row of data points.

CLC00004 Unable to evaluate linReg: coefficient is out of range.

Explanation:

The value that you specified for the nParam parameter of the linreg() function is invalid.

User response:

Specify either a 0 or a 1 for the nParam parameter. A 0 returns the slope of the linear regression. A 1 returns the intercept of the linear regression.

CLC00005 Illegal data row range specified: from {0} to {1}

Explanation:

The range of data rows specified for the nFirstRow and nLastRow parameters is invalid.

User response:

Specify a valid range of data rows. The nFirstRow value must specify a row before the nLastRow value in the query result set.

CLC00006 Unable to calculate gammaIP: domain error.

Explanation:

Invalid arguments were specified in the gammaIP function.

User response:

Specify a valid argument in the gammaIP function.

CLC00009 "{0}" function is not found in Lib class, but supported in expressions.

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00010 "{0}" function has wrong arguments description.

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00011 Type conversion error. Object of {0} type cannot be converted to {1} type.

Explanation:

The indicated object can not be converted to the indicated type.

User response:

Specify a valid object to be converted to the indicated type.

CLC00012 Evaluation of expression failed: {0}

Explanation:

The evaluation of the indicated expression failed.

User response:

Specify a valid expression for evaluation.

CLC00014 **Evaluation of expression has failed.**

Explanation:
The evaluation of the specified expression has failed.

User response:
Check the syntax of the expression for errors and try again.

CLC00016 **Cannot parse an expression: {0}**

Explanation:
An internal error occurred.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00018 **Unknown function name: {0}.**

Explanation:
The indicated function name is not valid

User response:
Specify a valid function name.

CLC00030 **Unable to process URL: "{0}"**

Explanation:
The indicated URL is invalid.

User response:
Specify a valid URL.

CLC00034 **"{0}" function description is missing.**

Explanation:
An internal error occurred.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00038 **Multiple declarations of "{0}" library in LibraryFactory class.**

Explanation:
An internal error occurred.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00039 **Multiple declarations of "{0}" library in LibraryFactory class.**

Explanation:
An internal error occurred.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00040 **Cannot load "{0}" library in LibraryFactory class.**

Explanation:
An internal error occurred.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00041 **Types "{0}" and "{1}" are not acceptable for "{2}" function.**

Explanation:
The arguments that were passed to the function are of invalid types.

User response:
Check the syntax of the expression for errors and try again.

CLC00042 **Type "{0}" is not acceptable for "{1}" function.**

Explanation:
The specified function does not accept parameters of the given type.

User response:
Check the syntax of the expression for errors and try again.

CLC00043 **Information service is undefined.**

Explanation:
An internal error occurred.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00044 **Aggregation service is undefined.**

Explanation:
An internal error occurred.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00045 **Hierarchy service is undefined.**

Explanation:
An internal error occurred.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00046 **Color service is undefined.**

Explanation:
An internal error occurred.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00047 **PPI service is undefined.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00048 **Value provider is undefined.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

CLC00049 **Function "{0}" is defined for {1} argument(s).**

Explanation:

An invalid number of parameters were specified for the indicated function.

User response:

Specify a correct number of parameters for the indicated function.

CLC00050 **Function "{0}" is undefined for argument {1}.**

Explanation:

An invalid number of parameters were specified for the indicated function.

User response:

Specify a valid number of parameters for the indicated function.

CLC00051 **One of the operands does not have measurement units.**

Explanation:

One of the operands of the function does not have measurement units.

User response:

Make sure that all of the operands in the function have measurement units.

CLC00052 **Cannot multiply two measured values.**

Explanation:

Measured values can not be multiplied.

User response:

Specify values that are not measured to be multiplied.

CLC00053 **Cannot create list from elements of "{0}" type.**

Explanation:

A list can not be created using elements of the indicated type.

User response:

Specify elements of a different type to be listed.

CLC00054 **List cannot contain elements of different types.**

Explanation:

The specified list can not contain elements of different types.

User response:

Specify elements of the same type to be listed.

CLC00055 **Types "{0}", "{1}" and "{2}" are not acceptable for "{3}" function.**

Explanation:

The indicated function is not compatible with the given types.

User response:

Check the syntax of the expression for errors and try again.

CLC00056 **Argument(s) with measurement units are not acceptable for "{0}" function.**

Explanation:

The indicated function can not contain arguments with measurement units.

User response:

Remove all measurement units from the arguments of the function.

CLC00057 **Incorrect index given.**

Explanation:

An error occurred in the LineText function.

User response:

Check the arguments that are specified for the LineText function for errors.

CLC00058 **Types "{0}", "{1}", "{2}" and "{3}" are not acceptable for "{4}" function.**

Explanation:

The indicated function is not compatible with the given types.

User response:

Check the expression syntax for errors and try again.

CLC00059 **Calculator exception cause: {0}**

Explanation:

A generic error occurred.

User response:

Check the syntax of the expression for errors and try again.

CLC00060 **Value for variable "{0}" is not defined.**

Explanation:

The value for the indicated variable is not defined.

User response:

Define the value for the indicated variable.

CLC00061 Function "{0}" caused error: {1}

Explanation:

The indicated function caused the indicated error.

User response:

Correct the indicated error and rerun the function.

CLC00062 Property "{0}" contains error: {1}

Explanation:

The indicated property caused the indicated error.

User response:

Correct the indicated error and rerun the function.

CLC00063 List sizes are not comparable.

Explanation:

Invalid parameters with type List were specified for add, multiply, divide, or subtract functions.

User response:

Specify a valid value for the List parameter of any add, multiply, divide, or subtract functions.

CLC00066 Cannot be evaluated for empty list.

Explanation:

No value was specified for the List parameter of a function.

User response:

Specify a valid value for the List parameter of any relevant function.

CLC00067 Function @if has odd number of parameters: @if(cond1, exp1, cond2, exp2, ..., expn). expn is used when there aren't true conditions

Explanation:

The syntax of an @if function is invalid.

User response:

Check the syntax of any @if functions for errors and try again.

DAL01001 The feature is not supported.

Explanation:

The specified feature is not supported by the database.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01002 No column named "{0}" was found.

Explanation:

The indicated column was not found.

User response:

Check the column name for errors or specify a different column name.

DAL01004 The current row is unsuitable for get/set operations.

Explanation:

The specified row can not be retrieved or set.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01005 The database cursor has been closed.

Explanation:

The requested operation could not be completed because the cursor is closed.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01006 An error occurred while accessing the file. {0}

Explanation:

An input/output error occurred while accessing the file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01007 Character encoding {0} is not supported.

Explanation:

The indicated character encoding is not supported.

User response:

Use a character set that is supported by your current JVM.

DAL01008 An error occurred while accessing the database.{0}

Explanation:

An error occurred while accessing the indicated database.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01009 An internal error occurred. {0}

Explanation:

The indicated internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01011 **Class {0} is not suitable for server description****Explanation:**

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01012 **Requested operation is not available in offline mode****Explanation:**

The specified operation is not available in offline mode.

User response:

Specify a different operation to apply to the active object, or switch to online mode and try again.

DAL01013 **Cannot create connection to server "{0}" due to the following: {1}****Explanation:**

A connection to the indicated server could not be established due to the indicated error.

User response:

Resolve the indicated error and retry to connect to the server.

DAL01014 **Invalid set of connection parameters used,****Explanation:**

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01016 **Cannot parse data (base64 encoding rules violation)****Explanation:**

The XML file is corrupted.

User response:

Fix any errors in the XML file and try again.

DAL01017 **Duplicate query with name "{0}" is detected in package "{1}".****Explanation:**

An internal error occurred; the indicated package contains several queries with the same specified name.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01018 **The source data is not compatible with the target table.****Explanation**

The attempt to save data in the database table failed because of the following:

- Number of columns does not match.
- Columns have incompatible types.

User response:

Select compatible target table and try again.

DAL01019 **Operation is not supported.****Explanation:**

An internal error occurred; the requested operation can not be performed due to internal limitations.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01020 **Invalid SQL type definition: "{0}"****Explanation:**

An internal error occurred; the indicated SQL type definition is invalid (it can be mapped to more than one known data types).

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01021 **This product does not contain SQLj support module****Explanation:**

QMF does not support or can not initialize the SQLj support module.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01022 **Cannot have two different driver libraries with the same driver name.****Explanation:**

The driver with the specified driver name is already defined in another library definition.

User response:

Use (edit) the already defined driver library instead of creating a new one.

DAL01023 **An error has occurred while saving driver libraries list: {0}****Explanation:**

The indicated error occurred while the driver libraries list was being saved.

User response:

Resolve the indicated error and try again.

DAL01025 **An error has occurred while loading the driver libraries list: {0}**

Explanation:

The indicated error occurred while trying to load the driver libraries list.

User response:

Resolve the indicated error and try again.

DAL01026 **Operation cannot be performed due to resource limits restriction: {0}**

Explanation:

The operation could not be performed due to the indicated resource limits restriction.

User response:

Check your current resource limits by selecting **Resource Limits** from the **View** menu. Specify an operation within your resource limits or contact your administrator for specific permissions.

DAL01027 **User name is missing.**

Explanation:

The user name is missing from the **User Information** dialog name field.

User response:

Specify a user name.

DAL01028 **Password is missing.**

Explanation:

The password is missing from the **User Information** dialog.

User response:

Specify a password.

DAL01029 **Wrong JDBC URL: {0}.**

Explanation:

The indicated JDBC URL cannot be processed by QMF. The URL may be incorrect.

User response:

Check the JDBC URL for errors or specify a different URL.

DAL01030 **Query parameter {0} is of wrong Java type. Data type "{1}" is required.**

Explanation:

An internal error occurred while running the query. The query parameters have invalid types.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01031 **Cached server descriptor cannot be restored due to the following: {0}**

Explanation:

The cached server descriptor cannot be restored due to the following error.

User response:

Resolve the indicated error and try again.

DAL01032 **An error occurred while making remote method call. {0}**

Explanation:

The indicated error occurred while the operation was running on the server side.

User response:

Refer to the indicated error description for possible response(s).

DAL01033 **Client request cannot be processed by remote server.**

Explanation:

An internal error occurred; the client request cannot be processed by a remote server.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01034 **Attempt to use incomplete metadata descriptor is detected.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01035 **Editing is not supported for this data set.**

Explanation:

An internal error occurred; this data set cannot be edited.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01036 **The current row is unsuitable for editing.**

Explanation:

An internal error occurred; the current row can not be edited.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01037 **The database server does not support table renaming**

Explanation:

Tables in this database server can not be renamed by QMF.

User response:
No action is required.

DAL01038 **The edit operation was cancelled.**

Explanation:
The table editing operation was cancelled by the user.

User response:
No action is required.

DAL01039 **The record could not be updated
(possibly concurrent edit/delete
operations).**

Explanation:
The record could not be updated. This is most likely because it is locked due to a concurrent edit/delete operation.

User response:
Refresh the result set and try again.

DAL01040 **The record could not be deleted
(possibly concurrent edit/delete
operations).**

Explanation:
The record could not be deleted. This is most likely because it is locked due to a concurrent edit/delete operation.

User response:
Refresh the result set and try again.

DAL01041 **The value of a calculated column
is longer than specified in the
metadata**

Explanation:
The data could not be exported because one or more of the calculated columns in the result set contains data that is longer than what is allowed in the column's definition.

User response:
Increase the maximum column length or modify the data in the calculated column.

DAL01042 **Invalid type conversion detected.**

Explanation:
An internal error occurred; a type conversion in the specified operation is invalid.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01043 **An attempt was made to modify a
deleted record.**

Explanation:
An internal error occurred; records that have already been deleted can not be modified or restored.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01044 **Connection is blocked due to the
following: {0}**

Explanation:
The attempted connection was blocked due to the indicated error or restriction.

User response:
Depending on the indicated problem, either resolve the specified error or contact your administrator to remove the restriction.

DAL01046 **This product does not contain
static mode support module for
selected database.**

Explanation:
This product does not contain static mode support module for selected database.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01048 **Driver class name "{0}" is not
loaded. This may be caused by
an incorrect driver class name or
missing JDBC driver libraries.**

Explanation:
The indicated driver class name can not be loaded. This may be caused by an incorrect driver class name or missing JDBC driver libraries.

User response:
Check that there is a proper definition for the JDBC driver library that refers to the indicated driver class.

DAL01049 **Selected protection mode is not
supported by database.**

Explanation:
The selected protection mode is not supported by the specified database.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01050 **Driver specific functionality is not
available due to: {0}**

Explanation:
Driver specific functionality is not available due to the indicated error.

User response:
Resolve the indicated error and retry. Verify that the JDBC driver library for the driver is properly configured and that it contains all of the necessary JAR files. If the problem persists, collect support data by selecting

Help > Collect Support Data and forward it to customer support.

DAL01200 **Generic data access error has occurred. {0}**

Explanation:

An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01201 **Generic data access error has occurred. {0}, {1}**

Explanation:

An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01202 **Generic data access error has occurred. {0}, {1}, {2}**

Explanation:

An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01203 **Generic data access error has occurred. {0}, {1}, {2}, {3}**

Explanation:

An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01204 **Generic data access error has occurred. {0}, {1}, {2}, {3}, {4}**

Explanation:

An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01205 **Generic data access error has occurred. {0}, {1}, {2}, {3}, {4}, {5}**

Explanation:

An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01206 **Generic data access error has occurred. {0}, {1}, {2}, {3}, {4}, {5}, {6}**

Explanation:

An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01207 **Generic data access error has occurred. {0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}**

Explanation:

An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DAL01208 **Generic data access error has occurred. {0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}**

Explanation:

An error with no textual description occurred. Additional information may follow the message. This error code may refer to different error conditions depending on product versions.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

DDM0001 **Unexpected error occurred.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0002 **Object {0} have no fields with codepoint {1}**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0003 **Socket open exception. {0}**

Explanation:

Unable to establish TCP/IP connection.

User response:

Check the host and port details in the JDBC URL; ping the server to check that the network connection works.

DDM0004 **Socket write exception. {0}**

Explanation:

There was a network connection failure or an internal error in the driver code was detected.

User response:

Check you network connection. If the connection works fine, send an error report to the developers.

DDM0005 **Data format exception detected.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0006 **Invalid JDBC URL {0}**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0007 **DSS header format error detected.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0008 **Unsupported CCSID {0}**

Explanation:

An internal error was detected in the driver code.

User response:

Send an error report to the developers.

DDM0009 **Cipher error {0}**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0010 **Object {0} does not contain required field {1}**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0011 **Object {0} does not support value {1}**

Explanation:

An internal error was detected in the driver code.

User response:

Send an error report to the developers.

DDM0012 **Syntax error detected: {0}**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0013 **Syntax error detected: {0}**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0014 **Object {0} can hold no more repeated fields {1}**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0015 **Illegal port number value: {0}**

Explanation:

The JDBC URL contains an invalid value in the **Port** field.

User response:

Specify a correct port number in the JDBC URL.

DDM0016 **{0} is invalid length for the field {1} of object {2}**

Explanation:

An internal error was detected in the driver code.

User response:

Send an error report to the developers.

DDM0017 **Unknown FD:OCA descriptor type id {0}**

Explanation:

An internal error was detected in the driver code.

User response:

Send an error report to the developers.

DDM0018 **Null value for non nullable data detected.**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0019 **Whole number of FD:OCA {0} triplet repeating groups cannot be placed in {1} bytes**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0020 **Illegal CPT triplet detected.**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0021 **Illegal duplicated field {1} in object {0}**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0022 **Referenced FD:OCA triplet {0} not found**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0023 **FD:OCA {0} triplets cannot be referenced from RLO triplet**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0024 **Replication factor {0} is detected in RLO triplet while only 1 and 0 is allowed by DRDA**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0025 **FD:OCA {0} triplets cannot be referenced from GDA triplet**

Explanation:

An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0026 **Unknown DRDA type {0}**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0027 **Infinite element count is not allowed for RLO handler of {0}**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0028 **An error occurred while parsing triplet handler for DRDA type {0}**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0029 **Unspecified value for JDBC statement variable {0}**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0030 **Illegal type conversion.**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0031 **Illegal date or time format detected: {0}**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0032 **Prepared statement parameter type {0} (from java.sql.Type set) is not supported.**

Explanation:
An internal error was detected in the driver code.

User response:
Send an error report to the developers.

DDM0033 **Unknown Db2 type {0}**

Explanation:

An internal error was detected in the driver code.

User response:

Send an error report to the developers.

DDM0035 Statement is closed.

Explanation:

An attempt to work with a closed Statement object was detected.

User response:

Fix your program code.

DDM0036 ResultSet is closed.

Explanation:

An attempt to work with a closed ResultSet object was detected.

User response:

Fix your program code.

DDM0037 Object is closed.

Explanation:

An attempt to work with a closed object was detected.

User response:

Fix your program code.

DDM0038 Operation cannot be performed in current ResultSet state.

Explanation:

An attempt to work with a ResultSet column of a result set that is not positioned on any valid record was detected.

User response:

Fix your program code.

DDM0039 Error reply is received.

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0040 Error reply is received for {0} command.

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DDM0041 Error detected while processing {0} object.

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DRL00001 The maximum number of connections to the server has been reached.

Explanation:

The maximum number of connections to the server has been reached.

User response:

Increase the **maximum of number of connections** parameter in the Repository Storage dialog or set the value to zero to remove this restriction.

DSCERRCD000 1 FD:OCA triplet is not used in DRDA descriptors, or the type code is invalid.

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD000 2 FD:OCA triplet sequence error.

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD000 3 An array description is required, and this is not one (too many or too few Row Lay Out (RLO) triplets).

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD000 4 A row description is required, and this is not one (too many or too few RLO triplets).

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD000 5 Late environmental descriptor just received, not supported.

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD000 6 Malformed triplet; required parameter is missing.

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD000 **Parameter value is not acceptable.**
7

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD001 **Meta-Data Descriptor (MDD)**
7 **present is not recognized as a**
 Structured Query Language (SQL)
 descriptor.

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD001 **MDD class is not recognized as a**
8 **valid SQL class.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD001 **MDD type is not recognized as a**
9 **valid SQL type.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD003 **Representation is incompatible**
3 **with SQL type (in prior MDD).**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD003 **CCSID is not supported.**
4

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD005 **Group Data Array (GDA) references**
0 **a local identifier (LID) which is not**
 a Scalar Data Array (SDA) or GDA.

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD005 **GDA length override exceeds**
1 **limits.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD005 **GDA precision exceeds limits.**
2

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD005 **GDA scale greater than precision**
3 **or scale negative.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD005 **GDA length override missing or**
4 **incompatible with data type.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD006 **RLO references a LID which is not**
5 **an RLO or GDA.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

DSCERRCD006 **RLO fails to reference a required**
6 **GDA or RLO.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

ERM0001 **Permanent Agent Error.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

ERM0002 **Not Authorized to Command.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

ERM0003 **Command Check.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0004 **Command Not Supported.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0005 **Manager Dependency Error.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0006 **Manager-level Conflict.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0007 **Conversational Protocol Error.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0008 **Parameter Not Supported.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0009 **Resource Limits Reached.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0010 **Data Stream Syntax Error.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0011 **Target Not Supported.**

Explanation:
An internal error in the driver code was detected.

User response:

Send an error report to the developers.

ERM0012 **Parameter Value Not Supported.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0013 **Conversational Protocol Error.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0014 **RDB Currently Accessed.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0015 **RDB Access Failed Reply Message.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0016 **Not Authorized To RDB.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0017 **RDB Not Found.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0018 **Resource Limits Reached.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0019 **Data Stream Syntax Error.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0020 **Invalid Request.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0021 **Abnormal End Unit of Work Condition.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0022 **RDB Package Binding Process Active.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0023 **Query Not Open.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0024 **RDB Not Accessed.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0025 **Data Descriptor Mismatch.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0026 **Object Not Supported.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0027 **RDB Package Binding Not Active.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0028 **RDB Update Reply Message.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0029 **SQL Error Condition.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0030 **End of Data.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0031 **End of Query.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0032 **Commitment Request.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0033 **End Unit of Work Condition.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0034 **Invalid Descriptor.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0035 **Open Query Complete.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0036 **Open Query Failure.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0037 **Query Previously Opened.**

Explanation:
An internal error in the driver code was detected.

User response:
Send an error report to the developers.

ERM0038 **Command Violation.**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

ERM1200 **Generic DRDA error has occurred.**
{0}

Explanation:

The indicated generic DRDA error has occurred.

User response:

Resolve the indicated generic DRDA error and try again.

ERM1201 **Generic DRDA error has occurred.**
{0}, {1}

Explanation:

The two indicated generic DRDA error have occurred.

User response:

Resolve the two indicated errors and try again.

ERM1202 **Generic DRDA error has occurred.**
{0}, {1}, {2}

Explanation:

The three indicated generic DRDA errors have occurred.

User response:

Resolve the three indicated errors and try again.

ERM1203 **Generic DRDA error has occurred.**
{0}, {1}, {2}, {3}

Explanation:

The four indicated generic DRDA errors have occurred.

User response:

Resolve the four indicated errors and try again.

ERM1204 **Generic DRDA error has occurred.**
{0}, {1}, {2}, {3}, {4}

Explanation:

The five indicated generic DRDA errors have occurred.

User response:

Resolve the five indicated errors and try again.

ERM1205 **Generic DRDA error has occurred.**
{0}, {1}, {2}, {3}, {4}, {5}

Explanation:

The six indicated generic DRDA errors have occurred.

User response:

Resolve the six indicated errors and try again.

ERM1206 **Generic DRDA error has occurred.**
{0}, {1}, {2}, {3}, {4}, {5}, {6}

Explanation:

The seven indicated generic DRDA errors have occurred.

User response:

Resolve the seven indicated errors and try again.

ERM1207 **Generic DRDA error has occurred.**
{0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}

Explanation:

The eight indicted generic DRDA errors have occurred.

User response:

Resolve the eight indicated errors and try again.

ERM1208 **Generic DRDA error has occurred.**
{0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}

Explanation:

The nine indicated generic DRDA errors have occurred.

User response:

Resolve the nine indicated errors and try again.

EXPT1000 **Cannot perform export command:**
{0}

Explanation:

The indicated export command can not be performed.

User response:

Resolve the first indicated error and try again.

EXPT1001 **An unsupported data type {0} was encountered.**

Explanation:

The result set contains columns with the types that are not supported by the IXF370 format.

User response:

Make sure that the result set does not contain columns with any of the following types: BINARY, VARBINARY, ROWID, LONGVARBINARY, CLOB, DBCLOB, BLOB, GRAPHIC, VARGRAPHIC, or LONGVARGRAPHIC.

EXPT1002 **The length of column {0} must be less than {1}.**

Explanation:

The length of the indicated column must be less than the indicated length due to restrictions of the IXF format.

User response:

Make sure that the indicated data column is less than the indicated length. Try to export using another format (for example, XML).

EXPT1003 **The precision of column {0} must be an odd number between {1} and {2}, inclusive.**

Explanation:

The precision of the indicated column must be an odd number between the two indicated values due to restrictions of the IXF format.

User response:

Make sure the precision of the indicated data column is an odd number between the two indicated values. Try to export using another format (for example, XML).

EXPT1004 **The data could not be exported due to an unsupported CCSID {0}.**

Explanation:

The indicated CCSID is not supported for export.

User response:

Specify a valid CCSID for export. A list of supported CCSIDs can be obtained from the **CCSID** drop down menu in the **Export** dialog.

EXPT1005 **The data could not be exported because the single-byte code page is not specified.**

Explanation:

A result set containing CLOB, CHAR, VARCHAR, or LONGVARCHAR columns was exported in IXF format but a CCSID that contains a single-byte code page was not specified.

User response:

Select a CCSID that contains a single-byte code page.

EXPT1006 **The data could not be exported because the double-byte code page is not specified.**

Explanation:

A result set containing DBCLOB, GRAPHIC, VARGRAPHIC, or LONGVARGRAPHIC columns was exported in IXF format but a CCSID that contains a double-byte code page was not specified.

User response:

Select a CCSID that contains a double-byte code page.

EXPT1007 **The value in column {0} exceeds the maximum length supported by the dBASE file format.**

Explanation:

The value in the indicated column exceeds the character limit that is supported by the dBASE file format.

User response:

Shorten the value in the indicated column or specify a different file format for export.

EXPT1008 **The data type of column {0} is not supported in dBASE files.**

Explanation:

The data type of the indicated column is not supported by the dBASE file format.

User response:

Specify a different data type for the indicated column or select a different file format for export. Supported types: CHAR, VARCHAR, SMALLINT,

INTEGER, DECIMAL, BIGINT, DOUBLE, FLOAT, DATE, and TIME.

EXPT1009 **The row length in a dBASE file cannot exceed 4000 characters.**

Explanation:

The summary length of all of the result set's columns exceeds 4,000 characters, which is not supported by the dBASE file format.

User response:

Make sure that the length is shorter than 4,000 characters, or select a different file format for export.

EXPT1010 **The specified data source does not exist.**

Explanation:

There was an attempt to export a result set to a data source that does not exist

User response:

Specify a valid data source.

EXPT1011 **The operation cancelled.**

Explanation:

The user cancelled the export command when prompted for a user name and password.

User response:

Specify a valid user name and password for the export command.

EXPT1012 **Name must be specified.**

Explanation:

There was an attempt to export a result set to a table that does not exist.

User response:

Specify a valid export table.

EXPT1013 **Invalid storage entry.**

Explanation:

The storage file contains corrupted data.

User response:

Setup fonts and font mapping for PDF export.

EXPT1014 **Cannot create font {0} for PDF**

Explanation:

The indicated font can not be created for the PDF file format.

User response:

Specify a different font for conversion, specify a different file format for export, or set up fonts and font mapping for PDF file export.

EXPT1015 **{0} font is not mapped**

Explanation:

The indicated font name is not mapped to the physical font file.

User response:

Set up fonts and font mapping for PDF file export.

FRM1001 The input line type is invalid.

Explanation:

Corrupted form. The form loader is hanging on a line with a certain first character, because every line in a form file, according to the form file format, starts with certain character (T, V, R, H, E, *) that define the line type. The current line's first character has an unknown type.

User response:

Try to open and save again with exact same version of QMF that was used to create the form.

FRM1002 The input line is too short.

Explanation:

The form loader is hanging on a line with a certain character count, because each line type must have a character count greater than a certain number (T > 15, V > 11, R > 3, or H > 44). The current line is too short.

User response:

Try to open and save again with exact same version of QMF that was used to create the form.

FRM1003 The boolean value is invalid. Valid values are YES or NO.

Explanation:

Corrupted form: A boolean value in the form file has a string other than YES or NO.

User response:

Try to open and save again with exact same version of QMF that was used to create the form.

FRM1004 The numeric value is invalid. Valid values are NONE, COLUMNS, DEFAULT, or an integer value.

Explanation:

Corrupted form: The form loader waits for a valid Integer value or predefined word (NONE, COLUMNS, or DEFAULT).

User response:

Try to open and save again with exact same version of QMF that was used to create the form.

FRM1005 An invalid code was encountered.

Explanation:

Corrupted form: Every table structure in a form file has its own column types. The form loader waits for certain column types depending on the table type.

User response:

Try to open and save again with exact same version of QMF that was used to create the form.

FRM1006 The object header is too short.

Explanation:

The form loader is hanging on a line with a certain character count, because each line type must have a character count greater than a certain number (T > 15, V > 11, R > 3, or H > 44). The current line is too short.

User response:

Try to open and save again with exact same version of QMF that was used to create the form.

FRM1007 The object header is invalid.

Explanation:

Corrupted form: The object header contains unexpected fields or data.

User response:

Try to open and save again with exact same version of QMF that was used to create the form.

FRM1008 Invalid usage code.

Explanation:

The "new column" of the form dialog contains an invalid usage code.

User response:

Leave the usage code field empty or select one from the combo box.

FRM1009 The value specified for {0} is invalid. Valid values are {1}.

Explanation:

In the form dialog page, there are several places where you can type or choose a listed value. An invalid listed value was entered.

User response:

Enter a valid value from the list.

FRM1010 The column heading "{0}" is too long.

Explanation:

The length of the column heading must be less than 40 characters.

User response:

Ensure that the column heading is less than 40 characters.

FRM1011 There is no condition expression with ID {0}.

Explanation:

A form has a few detail blocks in the "detail" branch (by default – one) and each detail can use a condition that defines the detail as enabled or disabled. The condition typed in the C1, C2, .. C999 format and the condition with the according index should exist in the "conditions" branch.

User response:

Specify YES, NO, or Ci where Ci is the existing condition.

FRM1012 **Line Wrapping Width must be NONE when Number of Fixed Columns is not NONE.**

Explanation:

In the **Options** page of the form designer, when any number is defined in the **Number of Fixed Columns** field, the **Line Wrapping Width** field must be set to NONE.

User response:

Specify NONE in either the **Line Wrapping Width** field or the **Number of Fixed Columns** field

FRM1013 **The ID for calculation {0} is already being used by a different calculation.**

Explanation:

In the **Calculations** page of the form designer, two or more calculations with the same ID were specified.

User response:

Ensure that each calculation ID is unique.

FRM1014 **The ID for condition {0} is already being used by a different condition.**

Explanation:

In the **Conditions** page of the form designer, two or more conditions with the same ID were specified.

User response:

Ensure that each of the conditions are unique.

FRM1015 **Line Wrapping Width must be 0 if any column uses a column wrapping edit code.**

Explanation:

If a column uses the wrapping edit code, then in the **Options** page of the form designer, the **Line Wrapping Width** field must be set to NONE.

User response:

Either set the **Line Wrapping Width** field to NONE, or do not use the line wrapping edit code.

FRM1016 **Automatically Reorder Report Columns must be TRUE when using the ACROSS usage code.**

Explanation:

If any columns have the ACROSS usage code, then in the **Options** page of the form designer, the **Automatically Reorder Report Columns** field must be set to TRUE.

User response:

Either set the **Automatically Reorder Report Columns** field to TRUE or do not use the ACROSS usage code.

FRM1017 **The ACROSS usage code requires other columns to use the GROUP and aggregate usage codes.**

Explanation:

If any columns use the ACROSS usage code, then all other columns must use either the OMIT, GROUP, or AGGREGATION usage code.

User response:

Set the OMIT, GROUP, or AGGREGATION usage codes for all other columns.

FRM1018 **Only one column can use the ACROSS usage code.**

Explanation:

Only one column can use the ACROSS usage code.

User response:

Ensure that only one column uses the ACROSS usage code.

FRM1019 **The GROUP usage code requires other columns to use aggregate usage codes.**

Explanation:

When one column uses the GROUP usage code, all of the other columns must use the OMIT, GROUP, BREAK, or AGGREGATION usage codes.

User response:

Ensure that all of the other columns use the OMIT, GROUP, BREAK, or AGGREGATION usage codes.

FRM1020 **The GROUP usage code requires all other columns to use a non-blank usage code.**

Explanation:

When a column uses the GROUP usage code, all of the other columns must use the OMIT, GROUP, BREAK, or AGGREGATION usage codes.

User response:

Ensure that all of the other columns use the OMIT, GROUP, BREAK, or AGGREGATION usage codes.

FRM1021 **The aggregate usage code can only be used when the column has a numeric data type.**

Explanation:

The SUM, CSUM, AVERAGE, STDEV, PCT, TPCT, CPCT, and TCPCT usage codes can only be used with numeric columns.

User response:

Specify a different usage code than SUM, CSUM, AVERAGE, STDEV, PCT, TPCT, CPCT, or TCPCT for non-numeric columns.

FRM1022 **There is no calculation expression with ID {0}.**

Explanation:

In the Calculations page of the form designer, the calculation usage code ID must be specified.

User response:

Specify a calculation ID for the calculation usage code.

FRM1023 **"{0}" is not a valid edit code for a calculation.**

Explanation:

The specified edit code is not a valid calculation edit code.

User response:

Specify another calculation edit code.

FRM1024 **Invalid edit code.**

Explanation:

The specified edit code is invalid.

User response:

Check the reference section of the User help to find a valid edit code.

FRM1025 **Invalid string "{0}" is encountered. Parsing failed.**

Explanation:

The form is corrupted or needs to be opened with another locale because a string that represents a number value can not be parsed in the current locale.

User response:

Try to open and save the form with the same version of QMF that was used to create the form or try to change the locale and reopen.

FRM1026 **The variable "{0}" is invalid in this context.**

Explanation:

The specified variable name is invalid when used in the expression.

User response:

Specify a valid variable name for the expression.

FRM1027 **There is no column {0}.**

Explanation:

The expression contains a variable that refers to a column that does not exist.

User response:

Specify a valid column number in the expression variable.

FRM1028 **The suffix '_B' is invalid in this context.**

Explanation:

The suffix '_B' can not be used in predefined variables, and as such is invalid for the current variable.

User response:

Remove the suffix '_B' from all predefined variables.

FRM1029 **Unable to open linked data source.**

Explanation:

The form attempted to open a linked data source and an error occurred during this process.

User response:

Specify a valid data source object for the form.

FRM1030 **An error occurred while registering user edit code.**

Explanation:

The FormCustomEditCode returned an error.

User response:

Review the FormCustomEditCode implementation documentation and check for errors.

FRM1031 **The user-defined edit routine class could not be found. {0}**

Explanation:

The user implementation class FormCustomEditCode can not be loaded by the Classloader.

User response:

Check the className.jar CLASSPATH.

FRM1032 **An error occurred while initializing user-defined edit routine class. {0}**

Explanation:

An error occurred.

User response:

See additional information in the wrapped exception.

GERR00001 **Generic driver error has occurred. {0}**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

GERR00002 **Generic driver error has occurred. {0}, {1}, {2}**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

GERR00003 **Generic driver error has occurred. {0}, {1}, {2}, {3}**

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

GERR00004 **Generic driver error has occurred.**
{0}, {1}, {2}, {3}, {4}

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

GERR00005 **Generic driver error has occurred.**
{0}, {1}, {2}, {3}, {4}, {5}

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

GERR00006 **Generic driver error has occurred.**
{0}, {1}, {2}, {3}, {4}, {5}, {6}

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

GERR00007 **Generic driver error has occurred.**
{0}, {1}, {2}, {3}, {4}, {5}, {6}, {7}

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

GERR00008 **Generic driver error has occurred.**
{0}, {1}, {2}, {3}, {4}, {5}, {6}, {7},
{8}

Explanation:

An internal error in the driver code was detected.

User response:

Send an error report to the developers.

IXF01002 **The LOB column index is invalid.**

Explanation:

The file contains an invalid index of a LOB column.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01003 **The LOB column length is invalid.**

Explanation:

The IXF file contains a LOB column that has an invalid length.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01004 **The end of the file was**
encountered prematurely.

Explanation:

A premature end of file was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01005 **The length of record {0} is invalid.**

Explanation:

A corrupted record was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01006 **An invalid record length was**
encountered in record {0}.

Explanation:

An invalid record length was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01007 **An invalid name length was**
encountered in record {0}.

Explanation:

An invalid column name length was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01008 **An unrecognized record type was**
encountered in record {0}.

Explanation:

An unknown record type was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01009 **An invalid value was encountered**
in record {0}.

Explanation:

An unexpected data value was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01010 **An unsupported data type was**
encountered in record {0}.

Explanation:

An unsupported data type was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01011 **An unsupported CCSID was encountered in record {0}.**

Explanation:

The IXF file contains character data which is encoded using a CCSID that is not supported by QMF.

User response:

Create an IXF file using another CCSID to import to QMF.

IXF01012 **The column length is invalid.**

Explanation:

An invalid column name was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01013 **The data record ID in record {0} is negative.**

Explanation:

An invalid data identifier was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01014 **The position for a column is negative in record {0}.**

Explanation:

An invalid column position was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01015 **The "IXF" identifier was not found in record {0}.**

Explanation:

The imported IXF file has no "IXF" signature, which is required at the beginning of its contents.

User response:

The imported IXF file is corrupted or has unknown format.

IXF01016 **The version of the IXF file is "{0}", which is not supported.**

Explanation:

The imported IXF file version is not supported by QMF.

User response:

Check to see if the IXF file is version 0, 1 or 2. all others are not supported.

IXF01017 **An invalid record count was encountered in record {0}.**

Explanation:

An invalid column count was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

IXF01018 **IXF file contents are corrupted.**

Explanation:

The imported IXF file is corrupted.

User response:

Check that you are specifying an IXF file to import.

IXF01019 **Data format is not supported in host IXF files.**

Explanation:

An unsupported column type was encountered while importing data from the IXF file.

User response:

The imported IXF file is corrupted or has an unknown format.

JOB1000 **{0}**

Explanation:

This is a wrapped message.

User response:

Resolve the indicated error and try again.

JOB1001 **Unknown JDF file format.**

Explanation:

An incorrect job file was selected while importing a job file in the old format (.jdf) to the new format (.jdfx).

User response:

Select a correct job file in the old format to import.

JOB1002 **Unexpected end of file.**

Explanation:

A corrupted job file was selected while importing a job file in the old format (.jdf) to the new format (.jdfx).

User response:

Resave the job file in QMF version 8 and retry to import the file.

LIC1000 **License field "{0}" has invalid date format: {1}.**

Explanation:

The format of the date at the specified field is incorrect.

User response:

Ensure that the license file contains a date in dd/mm/yyyy format.

LIC1001 **Required field "{0}" missed.**

Explanation:

No value was specified for the indicated field.

User response:

Ensure that a value is specified for the indicated field.

LIC1002 **Field "{0}" has invalid value "{1}".
Valid value is {2}.**

Explanation:

An incorrect value was specified for the indicated field.

User response:

Ensure that a valid license file is being used.

LIC1003 **License file could not be found {0}.**

Explanation:

The license file could not be found.

User response:

Use the **Manage License** dialog to add the license to the product.

LIC1004 **Unexpected file error: {0}.**

Explanation:

A file system error occurred while loading the license file.

User response:

Send an error report to the developers.

LIC1006 **License signature verification failed.**

Explanation:

The license file digital signature verification failed.

User response:

Provide a valid license file.

LIC1007 **License has expired.**

Explanation:

The trial period of the license expired.

User response:

Buy a full license.

LIC1008 **Unexpected error.**

Explanation:

An unexpected error occurred while verifying the license.

User response:

Ensure that the correct license file is being used.

LIC1009 **The license is incorrect.**

Explanation:

The license verification failed because of an invalid license file.

User response:

Use the original license file.

LIC1010 **License file not found.**

Explanation:

The license (any *.lic file) was not found the user's home directory or in application's directory.

User response:

Use the **Manage Licenses** dialog to import a valid license.

MAIL1000 **{0}**

Explanation:

The indicated error occurred.

User response:

Resolve the error and try again.

MAIL1001 **SMTP server must be specified.**

Explanation:

The email server that will send the message containing the object has not been specified in the procedure command.

User response:

Specify the email server that will send the message in the MAIL TO procedure command. Check the MAIL TO procedure command topic in the user help for more information.

MAIL1002 **Unknown type of attachment specified.**

Explanation:

The file specified in the MAIL TO command is of an unknown file format, or was entered incorrectly.

User response:

Check the ATTACHMENT parameter of the MAIL TO command for errors or specify a different attachment. Check the MAIL TO procedure command topic in the user help for more information.

NLS00001 **Generic error #{0} has occurred in module {1}.**

Explanation:

The indicated error occurred.

User response:

Send an error report to the developers.

NLS00002 **Generic error #{0} has occurred in module {1} ({2}).**

Explanation:

The indicated error occurred.

User response:

Send an error report to the developers.

NLS00003 **Generic error #{0} has occurred in module {1} ({2}; {3}).**

Explanation:
The indicated error occurred.

User response:
Send an error report to the developers.

NLS00004 **Generic error #{0} has occurred in module {1} ({2}; {3}; {4}).**

Explanation:
The indicated error occurred.

User response:
Send an error report to the developers.

NLS00005 **Generic error #{0} has occurred in module {1} ({2}; {3}; {4}; {5}).**

Explanation:
The indicated error occurred.

User response:
Send an error report to the developers.

NLS00006 **Generic error #{0} has occurred in module {1} ({2}; {3}; {4}; {5}; {6}).**

Explanation:
The indicated error occurred.

User response:
Send an error report to the developers.

NLS00007 **Generic error #{0} has occurred in module {1} ({2}; {3}; {4}; {5}; {6}; {7}).**

Explanation:
The indicated error occurred.

User response:
Send an error report to the developers.

NLS00008 **Generic error #{0} has occurred in module {1} ({2}; {3}; {4}; {5}; {6}; {7}; {8}).**

Explanation:
The indicated error occurred.

User response:
Send an error report to the developers.

NLS00009 **Generic error #{0} has occurred in module {1} ({2}; {3}; {4}; {5}; {6}; {7}; {8}; {9}).**

Explanation:
The indicated error occurred.

User response:
Send an error report to the developers.

OLCQ1000 **{0}**

Explanation:
This is a wrapped message.

User response:
Resolve the indicated error and try again.

OLCQ1001 **Query was cancelled.**

Explanation:
The current query was cancelled.

User response:
Rerun the query.

OLCQ1002 **You must select at least one value for each level.**

Explanation:
At least one value must be selected in every level of a slicer control.

User response:
Select a value and try again.

OLCQ1005 **Cannot to find needed data source.**

Explanation:
A nonexistent data source was specified for the OLAP query.

User response:
Specify a valid data source for the OLAP query.

OLP1000 **{0}**

Explanation:
This is a wrapped message.

User response:
Resolve the indicated error and try again.

OLP1002 **The database does not support OLAP feature.**

Explanation:
The specified database does not support OLAP functionality.

User response:
Specify a database with OLAP functionality.

OLP1003 **Unable to locate cube measure {0}.**

Explanation:
The indicated cube measure could not be found on the database.

User response:
Select a different database to search for the cube measure.

OLP1004 **Unable to locate cube dimension {0}.**

Explanation:
The indicated cube dimension could not be found on the database.

User response:

Select a different database to search for the cube dimension.

OLP1005 Unable to locate {0}.

Explanation:

The indicated OLAP cube could not be found on database.

User response:

Select a different database to search for the OLAP cube.

OLP1006 Cannot retrieve values for {0}.

Explanation:

An error occurred while reading the dimension information.

User response:

Ensure that the indicated dimension exists.

OLP1007 Measure or dimension cannot be added because it is not related to any table currently used in the layout.

Explanation:

An error occurred while building the SQL query to retrieve OLAP data from CubeViews; the table that stores information about the selected measure or dimension could not be found.

User response:

Check the integrity of the CubeView metadata.

OLP1008 There is not enough information to link specified measure or dimension: {0}.

Explanation:

An error occurred while building the SQL query to retrieve OLAP data from CubeViews; the tables that store information about the selected measures or dimensions could not be joined.

User response:

Check the integrity of the CubeView metadata.

OLP1009 Cannot connect to database: {0}.

Explanation:

Can not connect to the database.

User response:

Check if connection to a database is available or ask your administrator for more information.

OLP1010 Cube definition is not suitable for layout of the current query.

Explanation:

The cube structure as it is defined in the OLAP query does not match the current cube structure.

User response:

Check to make sure that the cube structure that is defined in the OLAP query matches the current cube structure.

OLQ1000 {0}

Explanation:

This is a wrapped message.

User response:

Resolve the indicated error and try again.

OLQ1001 Internal Error.

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

OLQ1002 Query was cancelled.

Explanation:

The current query was cancelled.

User response:

Rerun the query.

OLQ1003 You must select at least one value for each level.

Explanation:

At least one value must be selected in a slicer control.

User response:

Select a value and try again.

OLQ1004 Invalid type for saving the query.

Explanation:

An invalid object type was selected in which to save the query. OLAP queries can only be saved in the *.oq file format.

User response:

Save the OLAP query to a file with the *.oq file extension.

OLQ1005 No data source was specified for the query.

Explanation:

A nonexistent data source was specified for the OLAP query.

User response:

Specify a valid data source for the OLAP query.

OLQ1006 Only one reference to dimension hierarchy {0} can appear in the slicer.

Explanation:

MDX DB2 DWE supports only one dimension per slicer.

User response:

Select only one dimension.

PE00001 **Encountered "{0}" at column {1}.
Was expecting one of: {2}.**

Explanation:

An invalid character was encountered at the indicated column.

User response:

Replace the invalid character with one of the indicated characters.

PE00002 **Encountered "{0}" at column {1}.
Was expecting: {2}.**

Explanation:

An invalid character was encountered at the indicated column.

User response:

Replace the invalid character with the indicated character.

PE00003 **Expression needs a calculator to
get a value.**

Explanation:

The expression does not return a value.

User response:

No action is required.

PE00004 **Parse exception occurred.**

Explanation:

An error occurred while parsing the expression.

User response:

Check the expression syntax for errors and try again.

PRC1000 **An unexpected error occurred
while parsing the command.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

PRC1001 **An error occurred while parsing
the command: unexpected token
"{0}".**

Explanation:

An unexpected token was encountered while parsing the procedure.

User response:

Check the procedure's text for syntax errors (using the documentation or Content Assist via **Ctrl+Space**)

PRC1002 **An error occurred while parsing
the command: unexpected
character "{0}".**

Explanation:

An unexpected character was encountered while parsing the procedure.

User response:

Check the procedure's text for syntax errors or invalid characters (using the documentation or Content Assist via **Ctrl+Space**)

PRC1003 **The parameter "{0}" is invalid in
this context.**

Explanation:

An unsupported parameter token was encountered while parsing the procedure.

User response:

Check the procedure's text for syntax errors (using the documentation or Content Assist via **Ctrl+Space**)

PRC1004 **The parameter "{0}" cannot be
specified twice.**

Explanation:

The indicated parameter was specified twice in the same procedure command.

User response:

Only specify one instance of the indicated parameter.

PRC1005 **"{1}" is not a valid value for
parameter "{0}".**

Explanation:

The indicated value is not valid for the indicated parameter.

User response:

Specify a valid value for the indicated parameter. See the Procedure Commands section of the user help for more information on procedure commands and their parameters.

PRC1006 **You cannot set variable "{0}".**

Explanation:

The indicated variable can not be set manually.

User response:

Specify a valid variable and try again.

PRC1008 **An error occurred while parsing
the command: unexpected end of
line.**

Explanation:

An unexpected end of line was encountered while parsing the procedure.

User response:

Check the procedure's text for syntax errors (using the documentation or Content Assist via **Ctrl+Space**)

PRC1009 **An error occurred while parsing
the command: unrecognized
command "{0}".**

Explanation:

The indicated command is not valid.

User response:

Check the procedure text for errors or specify a different command.

PRC1010 **Parsing error at lines {0}--{1} (command #{2}). {3}**

Explanation:

a syntax error was encountered while parsing procedure.

User response:

Check the procedure's text for syntax errors (using the documentation or Content Assist via **Ctrl+Space**)

PRC1011 **An error occurred while parsing the command: invalid object name "{0}".**

Explanation:

The indicated object name is invalid.

User response:

Check the procedure text for errors and specify a valid object name. See the Procedure Commands section of the user help for more information.

PRC1012 **You must specify object type or object name.**

Explanation:

A required object type or name was not specified.

User response:

Specify a valid object type or name where needed in the procedure command text. See the Procedure Commands section of the user help for more information.

PRC1013 **"{0}" is an invalid variable name.**

Explanation:

The indicated variable name is invalid.

User response:

Check the variable name for syntax errors or specify another valid variable. Check the Procedure Commands section of the user help for more information.

PRC1014 **Close bracket reached before end.**

Explanation:

An unexpected close bracket was encountered while parsing the procedure.

User response:

Check the procedure's text for syntax errors (using the documentation or Content Assist via **Ctrl+Space**)

PRC2001 **An error occurred while retrieving {0}. {1}**

Explanation:

The indicated error occurred while retrieving the indicated object.

User response:

Resolve the error and rerun the procedure.

PRC2002 **{0} does not exist.**

Explanation:

The indicated object does not exist.

User response:

Specify a valid object in the procedure command.

PRC2003 **{0} exists but is of the wrong type.**

Explanation:

The indicated object exists in an open window of the editor, but the object type that was specified in the procedure command does not match the type of the object.

User response:

Make sure the object type that is specified in the procedure command matches the type of the object in the open window of the editor.

PRC2004 **{0} does not exist or is of the wrong type.**

Explanation:

There is no object of the specified object type open in an open window of the editor, or the object type is incorrect.

User response:

Make sure that the object type that is specified in the procedure command matches the type of the object that is in the open window.

PRC2005 **The procedure issued a command against a current object, but no current object of the correct type exists.**

Explanation:

The procedure does not specify a specific object name, only an object type. In this instance, if an object of this type is open in another window of the editor, the procedure will run correctly. Since an object of the type specified is not open in another window, the procedure failed to run.

User response:

Make sure an object of the specified type is open in another window, or specify a specific object name.

PRC2006 **{0}**

Explanation:

This is a wrapped message.

User response:

Resolve the indicated error and try again.

PRC2007 **The CONVERT command can only be issued against queries; it cannot be applied to {0}.**

Explanation:

The indicated object can not be converted using the CONVERT command.

User response:

Specify a query to be converted using the CONVERT command. For more information, see the CONVERT topic in the Procedure Commands section of the user help.

PRC2008 **The operation was cancelled.**

Explanation:

The operation was cancelled by the user.

User response:

Rerun the operation.

PRC2011 **Saving in {0} format is not supported.**

Explanation:

The selected format is not supported when exporting an object. Reports may only be exported in TEXT, HTML and PDF formats.

User response:

Change the export data format in the procedure command.

PRC2017 **The maximum nesting depth of procedures was exceeded.**

Explanation:

More than 50 nested procedure levels are not support.

User response:

Redesign the procedure to decrease its depth.

PRC2018 **The specified query did not return a result set.**

Explanation:

The SAVE DATA command was issued against a query that does not contain a result set.

User response:

Ensure that the query you are trying to save contains a result set.

PRC2020 **An error occurred while saving the object at the server. {0}**

Explanation:

The indicated error occurred while the object was being saved at the server.

User response:

Resolve the indicated error and rerun the procedure.

PRC2022 **The operation cannot be performed because no data is available.**

Explanation:

The EXPORT DATA command was issued against an object that does not contain data.

User response:

Ensure that the object that you are trying to export contains data.

PRC2025 **The procedure was terminated because it recursively invokes itself.**

Explanation:

The procedure recursively invoked itself. Recursion is not supported in procedures.

User response:

Rewrite the procedure and remove the recursion.

PRC2026 **An error occurred while running the procedure. {0}**

Explanation:

The indicated error occurred while the procedure was being run.

User response:

Resolve the indicated error and rerun the procedure.

PRC2028 **An error occurred while importing the {0} file.**

Explanation:

An error occurred while the file was being imported.

User response:

Ensure that object that is being imported is of a supported format.

PRC2029 **Could not connect to a server {1}: {0}**

Explanation:

The data source alias that was specified in the CONNECT TO command is undefined.

User response:

Ensure that the command refers to a valid data source in the repository.

PRC2030 **Specified object {0} for import has incompatible type.**

Explanation:

The IMPORT or DISPLAY command was issued against an object with an unsupported type.

User response:

Ensure that the object to be imported or displayed is of a valid type.

PRC2031 **Specified command, or some of its parameters unsupported.**

Explanation:

The procedure contains commands or parameters that are not supported.

User response:

Check the procedure's text for syntax errors (using the documentation or Content Assist via **Ctrl+Space**)

PRC2032 **Cannot save non-QMF object into QMF Catalog.**

Explanation:

The specified object is a non-QMF object and can not be saved into the QMF Catalog.

User response:

Specify a different location in which to save the object or specify a different object to be saved.

PRC2033 **You cannot set the value of {0}.**

Explanation:

The indicated parameter's value can not be manually set with the SET GLOBAL command.

User response:

Values for some DSQ variables cannot be set manually (for example DSQAO_CONNECT_ID, DSQAO_NUM_FETCHED, etc). Rewrite the procedure to exclude these variable values.

PRC2035 **The printer name "{0}" is invalid.**

Explanation:

The indicated printer name is invalid.

User response:

Check the spelling of the printer name, or specify a different printer to use for the PRINT command.

PRC2036 **You are not authorized to use the table editor at this data source.**

Explanation:

The **Enable Table Editing** resource limit for the data source is turned off; you can not open Table Editor of the data source

User response:

Ask your administrator to enable this resource limit.

PRC2037 **You are not authorized to edit tables at this data source.**

Explanation:

The **Enable Table Editing** resource limit for the data source is turned off; you cannot edit values in a query's grid cells.

User response:

Ask your administrator to enable this resource limit.

PROC1000 **REXX procedure references cannot be determined.**

Explanation:

References for the procedure cannot be built for REXX procedures.

User response:

You can view references at the **Repository** or **Workspaces** views for generic procedures only.

PROC1001 **Some substitution variables are not defined.**

Explanation:

Some of the substitution variables of the procedure are not defined.

User response:

Ensure that all of the substitution variables are defined for the procedure, or run this procedure without the / batch parameter in the command line. In this case QMF will ask for all of the necessary variables values.

QEL00001 **You are not allowed to run queries returning LOB data.**

Explanation:

You are not allowed to run queries that contain large object data.

User response:

Check your LOB Options resource limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QEL00002 **The Maximum Rows to Fetch limit has been exceeded. {0} rows were retrieved so far.**

Explanation:

The Maximum Rows to Fetch limit has been reached. The indicated number of rows have been retrieved so far.

User response:

Check your Maximum Rows to Fetch resource limit by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QEL00003 **The Maximum Bytes to Fetch limit has been exceeded. {0} bytes were retrieved so far.**

Explanation:

The Maximum Bytes to Fetch limit has been reached. The indicated number of bytes have been retrieved so far.

User response:

Check your Maximum Bytes to Fetch resource limit by selecting **Resource Limits** from the **View** menu to see

your specific allowances or ask your administrator for more information.

QEL00004 **The length of a LOB column exceeds the maximum allowable length.**

Explanation:

The length of a LOB column in the query results exceeds the maximum allowable length.

User response:

Check your Maximum LOB column size by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QEL00005 **You are not allowed to use the "{0}" SQL verb.**

Explanation:

You are not allowed to use the indicated SQL verb.

User response:

Check your SQL Verbs limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QEL00006 **You are not allowed to save data to database.**

Explanation:

You are not allowed to save data to the database.

User response:

Check you Save Data resource limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QEN00001 **Query was cancelled.**

Explanation:

The active query was cancelled.

User response:

Rerun the query.

QEN00003 **Unexpected token: {0}**

Explanation:

An unexpected token was encountered while parsing the query that calls the stored procedure.

User response:

Check the query's syntax for errors.

QEN00004 **Unexpected token: {0} is expected but {1} was found.**

Explanation:

An unexpected closed bracket was encountered while parsing the query that calls the stored procedure.

User response:

Check the query's syntax for errors.

QEN00005 **Unexpected end of statement.**

Explanation:

An unexpected end of statement was encountered while parsing the query that calls the stored procedure.

User response:

Check the query's syntax for errors.

QEN00006 **Expression {0} is not a stored procedure.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QEN00007 **End of statement is expected but {0} was found**

Explanation:

Tokens were encountered after the end of statement while parsing the query that calls the stored procedure.

User response:

Check the query's syntax for errors.

QEN00008 **Unsupported parameter data type: {0}**

Explanation:

An unsupported parameter was encountered while parsing the query that calls the stored procedure.

User response:

Check the query's syntax for errors.

QEN00009 **{0} is invalid value for this type: {1}**

Explanation:

When prompted for a value for a stored procedure parameter, the user entered a value that is not valid for that parameter's type.

User response:

Specify a valid value for the parameter.

QMF1004 **You are not authorized to print objects from "{0}" data source.**

Explanation:

The indicated data source name has a restriction on printing for the current user.

User response:

Contact your administrator to check your specific permissions for printing.

QMF1005 **QMF Catalog table structure is obsolete. Catalog tables must be updated.**

Explanation:

The indicated an error occurred because the QMF Catalog has an obsolete structure.

User response:

Contact your administrator to upgrade the QMF Catalog.

QMF1006 **You are not authorized to save data to files from "{0}" data source.**

Explanation:

The indicated data source name has a restriction on saving data for the current user.

User response:

Contact your administrator to check your specific permissions for saving data.

QRL00001 **You are not authorized to access this data source in user interface mode.**

Explanation:

You are not authorized to access this data source in user interface mode.

User response:

Check your Options limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QRL00002 **You are not authorized to access this data source in automation mode.**

Explanation:

You are not authorized to access this data source in automation mode.

User response:

Check your Options limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QRL00003 **You are not authorized to run unsaved queries.**

Explanation:

You are not authorized to run unsaved queries.

User response:

Check your Options limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QRL00004 **You are not authorized to save data from this data source to a file.**

Explanation:

You are not authorized to save data from this data source to a file.

User response:

Check your Save Data limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QRL00005 **You are not authorized to save data from this data source to the database.**

Explanation:

You are not authorized to save data from this data source to the database.

User response:

Check your Save Data limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QRL00006 **You are not authorized to retrieve LOB data.**

Explanation:

You are not authorized to retrieve LOB data.

User response:

Check your LOB Options limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QRL00008 **No resource group for user {0} was found.**

Explanation:

No resource group for the indicated user was found.

User response:

Specify a valid resource group for the user.

QRL00009 **The operation cannot be performed because the processing of LOB data is disabled.**

Explanation:

The operation cannot be performed because the processing of LOB data is disabled.

User response:

Check you LOB Options limits by selecting **Resource Limits** from the **View** menu to see your specific allowances or ask your administrator for more information.

QRY0999 **{0}**

Explanation:

This is a wrapped message.

User response:

Resolve the indicated error and try again.

QRY1001 **Query was cancelled.**

Explanation:

The query was cancelled.

User response:
Rerun the query.

QRY1002 **An invalid header record was encountered.**

Explanation:
The query was loaded with errors.

User response:
Try to reload the query; otherwise the query is corrupted.

QRY1003 **No data source was specified for the query.**

Explanation:
No data source was specified for the query.

User response:
Specify a data source for the query.

QRY1004 **An unexpected "R" record was encountered: {0}.**

Explanation:
An error was encountered while loading a type R record from the prompted query file during storage of info in the prompted query structure.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1005 **An unexpected "T" record was encountered: {0}.**

Explanation:
An error was encountered while loading a type T record from the prompted query file during storage of info in the prompted query structure.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1006 **An invalid "T" record was encountered: {0}.**

Explanation:
An error was encountered while loading a type T record from the prompted query file during storage of info in the prompted query structure.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1007 **A "T" record with invalid column information was encountered: {0}.**

Explanation:
An error was encountered while loading a type T record from the prompted query file during storage of info in the prompted query structure.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1008 **An unrecognized table type was encountered: {0}.**

Explanation:
An error was encountered while loading a record from the prompted query file during storage of info in the prompted query structure.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1009 **An invalid value in an "R" record for table 1110 was encountered: {0}.**

Explanation:
An error was encountered while loading a type R record from the prompted query file during storage of info in the prompted query structure.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1010 **An invalid value in an "R" record for table 1150 was encountered: {0}.**

Explanation:
An error was encountered while loading a type R record from the prompted query file during storage of info in the prompted query structure.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1011 **An invalid value in an "R" record for table 1210 was encountered: {0}.**

Explanation:
An error was encountered while loading a type R record from the prompted query file during storage of info in the prompted query structure.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1012 **An invalid value in an "R" record for table 1310 was encountered: {0}.**

Explanation:
An error was encountered while loading a type R record from the prompted query file during storage of info in the prompted query structure.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1013 **An invalid value in an "R" record for table 1410 was encountered: {0}.**

Explanation:

An error was encountered while loading a type R record from the prompted query file during storage of info in the prompted query structure.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1014 **An out of sequence 'R' record for table 1310 was encountered.**

Explanation:

An error was encountered while loading a type R record from the prompted query file during storage of info in the prompted query structure.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1015 **An invalid "V" record was encountered: {0}.**

Explanation:

An error was encountered while loading a type V record from the prompted query file during storage of info in the prompted query structure.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1016 **There must be less than 16 tables in Prompted Query.**

Explanation:

There must be less than sixteen tables called from the Prompted Query editor.

User response:

Ensure that there are less than sixteen tables called from the Prompted Query editor and rerun the query.

QRY1017 **The same correlation ID was encountered for two or more tables.**

Explanation:

The PqTable object for the specified correlation ID already exists.

User response:

Select another table for the Prompted Query editor.

QRY1018 **An invalid table correlation ID was encountered.**

Explanation:

An error was encountered while retrieving a valid correlation ID.

User response:

Try to add the query to the Prompted Query editor again, or collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1019 **A column on which to join tables is in more than one of the tables in the query.**

Explanation:

A join column can only exist in one table when multiple tables are joined in a single query.

User response:

Ensure that any join columns in the query exist in only one table.

QRY1020 **A column on which to join tables is not in any of the tables in the query.**

Explanation:

A specified join column is not present in any of the tables in the query.

User response:

Check the join column name for errors, or specify a different join column.

QRY1021 **Type of join columns for joined tables mismatch.**

Explanation:

A columns that join two or more tables must all be of the same column type.

User response:

Ensure that all of the columns that make up a single join are of the same column type.

QRY1022 **The expression contains a syntax error: {0}**

Explanation:

The indicated expression contains a syntax error.

User response:

Correct the syntax error and rerun the query.

QRY1023 **An invalid connector for a row condition was encountered.**

Explanation:

An invalid connector for a row condition was encountered.

User response:

Check all row conditions for errors and rerun the query.

QRY1024 **An invalid expression type in a row condition was encountered.**

Explanation:

An invalid expression type in a row condition was encountered.

User response:

Check all expression types in the row conditions for errors and rerun the query.

QRY1025 **An invalid operator in a row condition was encountered.**

Explanation:

An invalid operator in a row condition was encountered.

User response:

Check all operators in the row conditions for errors and rerun the query.

QRY1026 **The selected relational operator is not valid for numeric data.**

Explanation:

An invalid operator in a row condition was encountered for numeric data.

User response:

Check all of the operators in the row conditions for errors and rerun the query.

QRY1027 **An invalid sort direction was encountered.**

Explanation:

An invalid sort direction was encountered. Valid directions are **Ascending** or **Descending**.

User response:

Check all instances where sort directions are applied and change them to either ascending or descending.

QRY1028 **The prompted query format is an earlier version, and contains join conditions that cannot be migrated to the current version. These join conditions have been deleted.**

Explanation:

A join condition can only appear with the higher of the two tables involved in the join. Also, a table cannot be joined with a preceding table if there is a third table between them with no join conditions.

User response:

Reordered elements of the array of joins to match the current order of the tables.

QRY1029 **The table {0} does not exist.**

Explanation:

The indicated table is called in the SQL code, but does not exist in the data source.

User response:

Check the table name for errors, or enter a different table to be called.

QRY1030 **The prompted query cannot be saved because a table name exceeds the maximum length allowed in the prompted query export format.**

Explanation:

The prompted query cannot be saved because a table name exceeds the maximum length allowed in the prompted query export format.

User response:

Decrease the length of the table name.

QRY1031 **The prompted query cannot be saved because a column name used in a join condition exceeds the maximum length allowed in the prompted query export format.**

Explanation:

The prompted query cannot be saved because a column name used in a join condition exceeds the maximum length allowed in the prompted query export format.

User response:

Decrease the length of the join column name.

QRY1032 **The prompted query cannot be saved because a column expression exceeds the maximum length allowed in the prompted query export format.**

Explanation:

The prompted query cannot be saved because a column expression exceeds the maximum length allowed in the prompted query export format.

User response:

Decrease the length of the column expression name.

QRY1033 **The prompted query cannot be saved because an expression in a row condition exceeds the maximum length allowed in the prompted query export format.**

Explanation:

The prompted query cannot be saved because an expression in a row condition exceeds the maximum length allowed in the prompted query export format.

User response:

Decrease the length of the row condition.

QRY1034 **An invalid record was encountered.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1035 **Only a SELECT query can be shown in the prompted view.**

Explanation:

Only a SELECT query can be shown in the **Prompted Query** editor.

User response:

Make sure that the specified query contains a SELECT statement.

QRY1036 **Only simple SELECT statements are supported.**

Explanation:

Only simple SELECT statements are supported.

User response:

Ensure that the specified query only contains simple SELECT statements.

QRY1037 **Unrecognized table reference.**

Explanation:

An error occurred while parsing the SQL text to construct the prompted query.

User response:

Check the SQL text for errors and try again.

QRY1038 **You cannot view a query that has subselects in the prompted view.**

Explanation:

Subselects are not supported for queries that are viewed in the **Prompted Query** editor.

User response:

Ensure that specified query does not contain subselects if it is to be viewed in the **Prompted Query** editor.

QRY1039 **Table locator references are not supported.**

Explanation:

An error occurred while parsing the SQL text to construct the prompted query.

User response:

Check the SQL text for errors and try again.

QRY1040 **Table functions are not supported.**

Explanation:

An error occurred while parsing the SQL text to construct the prompted query.

User response:

Check the SQL text for errors and try again.

QRY1041 **Only AND conditions are supported for joins in the prompted view.**

Explanation:

Only AND conditions are supported for joins in queries that are viewed in the **Prompted Query** editor.

User response:

Ensure that the specified query only contains AND conditions for joins if it is to be viewed in the **Prompted Query** editor.

QRY1042 **Quantified predicates are not supported.**

Explanation:

An error occurred while parsing the SQL text to construct the prompted query.

User response:

Check the SQL text for errors and try again.

QRY1043 **'Exists' predicates are not supported.**

Explanation:

An error occurred while parsing the SQL text to construct the prompted query.

User response:

Check the SQL text for errors and try again.

QRY1044 **Only simple LIKE statements are supported.**

Explanation:

Only simple LIKE statements are supported.

User response:

Ensure that the specified query only contains simple LIKE statements.

QRY1045 **Only simple row conditions are supported.**

Explanation:

Only simple row conditions are supported.

User response:

Ensure that the specified query only contains simple row conditions.

QRY1046 **Expression cannot be empty.**

Explanation:

An error occurred while constructing the query.

User response:

Check the query's syntax for errors and try again.

QRY1047 **Invalid type for saving the query.**

Explanation:

The type that was set for the query is invalid.

User response:

Specify a valid type for the query or collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

QRY1048 **You cannot run an empty query.**

Explanation:

The specified query did not contain any SQL code.

User response:

Populate the query with at least a SELECT statement and rerun.

QRY1049 **Some substitution variables are not defined.**

Explanation:

Some of the substitution variables in the query were not given values when the query was run.

User response:

Ensure that all of the substitution variables in the query are defined.

QRY1050 **You cannot view a query that uses row values in the predicate.**

Explanation:

An error occurred while parsing the SQL text to construct the prompted query.

User response:

Check the SQL text for errors and try again.

QRY1051 **'XMLExists' predicates are not supported.**

Explanation:

An error occurred while parsing the SQL text to construct the prompted query.

User response:

Check the SQL text for errors and try again.

QRY1052 **An invalid value: {0}.**

Explanation:

An error occurred while parsing the SQL text to construct the prompted query.

User response:

Check the SQL text for errors and try again.

QRY1053 **Only SELECT statements are supported in compound queries.**

Explanation:

Only SELECT statements are supported in compound queries.

User response:

Ensure that only SELECT statements are called in the specified compound query.

QRY1054 **Query content is not compatible with Prompted Query format.**

Explanation:

The query can not be saved in the prompted format.

User response:

Save the query in another format.

QRY1055 **Host variable predicates are not supported.**

Explanation:

An error occurred while parsing the SQL text to construct the prompted query.

User response:

Check the SQL text for errors and try again.

QRY1540 **Selected column is a large object (LOB). Union operation is not supported for LOB data types.**

Explanation:

An error occurred when performing union transformation on a LOB column.

User response:

Do not perform union transformation on LOB data types.

QRY1541 **Selected column is a large object (LOB). Intersect operation is not supported for LOB data types.**

Explanation:

An error occurred when performing intersect transformation on a LOB column.

User response:

Do not perform intersect transformation on LOB data types.

QRY1542 **Selected column is a large object (LOB). Except operation is not supported for LOB data types.**

Explanation:

An error occurred when performing except transformation on a LOB column.

User response:

Do not perform except transformation on LOB data types.

RC00000 **Internal error**

Explanation:

An internal processor error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RC00001 **Type mismatch**

Explanation:

An unexpected error occurred while processing the *.rco file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RC00002 Function {0} is not implemented yet

Explanation:

An unexpected error occurred while processing the *.rco file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RC00003 Undefined variable {0}

Explanation:

An unexpected error occurred while processing the *.rco file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RC00004 Missing variable name

Explanation:

An unexpected error occurred while processing the *.rco file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RC00005 Syntax error in "{0}"

Explanation:

An unexpected error occurred while processing the *.rco file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RC00006 Unknown function {0}

Explanation:

An unexpected error occurred while processing the *.rco file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RC00007 Unknown object type {0}

Explanation:

An unexpected error occurred while processing the *.rco file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RC00008 Wrong parameters in "{0}"

Explanation:

An unexpected error occurred while processing the *.rco file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RC00009 Database object {0} not found

Explanation:

An unexpected error occurred while processing the *.rco file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RC00010 Database error {0}

Explanation:

An unexpected error occurred while processing the *.rco file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RC00011 Unknown database server type {0}

Explanation:

An unexpected error occurred while processing the *.rco file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

REP01000 {0}

Explanation:

The indicated error occurred while accessing the repository.

User response:

Resolve the indicated error and try again.

REP01001 An input/output error occurred: {0}

Explanation:

The indicated error occurred in the repository.

User response:

Resolve the indicated error and try again.

REP01002 Character encoding is not supported: {0}

Explanation:

The indicated encoding error occurred while the data was being processed.

User response:

Ensure that UTF-8 encoding is used in the repository storage database.

REP01003 Internal error has occurred: {0}

Explanation:

The indicated error occurred.

User response:

Resolve the indicated error and try again.

REP01004 The following error has occurred while accessing database: {0}

Explanation:

The indicated error occurred while the database was being accessed.

User response:

Resolve the indicated error and retry to connect to the database.

REP01005 Operation is not supported.

Explanation:

The specified operation is not supported by the repository.

User response:

Specify a different operation and try again.

REP01006 Object with such identifier not found.

Explanation:

An error occurred while searching for the object.

User response:

Ensure that the specified object identifier is correct.

REP01007 Object or file with URL "{0}" not found.

Explanation:

There are no objects or files with the indicated URL.

User response:

Check the URL for errors or specify a different URL.

REP01008 Not enough permissions to perform operation.

Explanation:

Your user ID does not have permission to perform the specified operation.

User response:

Contact your administrator to check your specific permissions or specify a different operation to perform.

REP01009 Operation cancelled due to invalid object identifier: {0}

Explanation:

The operation was cancelled because the indicated object identifier is invalid.

User response:

Specify a different object identifier and retry the specified operation.

REP01010 Cannot open connection: {0}

Explanation:

The indicated error occurred while establishing a connection to the repository database.

User response:

Ensure that the repository database is accessible.

REP01011 At least one line of the document is longer than 79 characters, which is the maximum supported length when saving objects in the database.

Explanation:

The maximum character length allowed for documents saved to the database is seventy-nine characters. The document contains at least one line that is longer than seventy-nine characters.

User response:

Ensure that all of the lines of the document are less than seventy-nine characters and retry to save the document.

REP01012 {0} already exists and has a different type.

Explanation:

The indicated object name already exists as a different type of object. Each object in the database must have a different name.

User response:

Specify a different name for the object and retry to save the object.

REP01013 {0} does not exist.

Explanation:

The indicated object could not be found because it does not exist in the database.

User response:

Specify a different object name to search for in the database.

REP01014 {0} already exists.

Explanation:

The indicated object could not be saved because it already exists in the database.

User response:

Specify a different name for the object and retry to save the object.

REP01015 {0} has unsupported type and cannot be accessed.

Explanation:

The indicated object is of an unsupported type.

User response:

Ensure that all of the necessary plug-ins are installed and available.

| | | |
|-----------------------|---|--|
| REP01016 | Operation is impossible because object is too large. | Check the repository initialization parameters and try again. |
| Explanation: | The specified operation could not be performed because the specified object is too large. | |
| User response: | Specify a different object or a different operation and try again. | |
| REP01017 | Illegal state or illegal operation error. | REP01027 A repository instance hasn't been initialized. |
| Explanation: | An internal error occurred in the repository. | Explanation: An error occurred while trying to access a non-initialized repository. |
| User response: | Collect support data by selecting Help > Collect Support Data and forward it to customer support. | User response: Initialize the repository and try again. |
| REP01018 | User is not logged on repository. | REP01028 Repository "{0}" could not be found. |
| Explanation: | You can not perform the specified operation, because you are not connected to the repository. | Explanation: The indicated repository could not be found. |
| User response: | Connect to the repository and try again. | User response: Check the repository name for errors or specify a different repository and try again. |
| REP01019 | Not implemented. | REP01029 Object type {0} cannot be added to this folder. |
| Explanation: | The indicated error occurred while attempting to access a non-implemented functionality. | Explanation: Objects of the indicated type can not be added to the specified folder. |
| User response: | Ensure that you are using the latest version of the product. | User response: Specify a different folder in which to add the indicated object. |
| REP01020 | Unexpected error occurred. | REP01030 More authentication information for {0} required. |
| Explanation: | An internal error occurred in the repository. | Explanation: Not enough authentication information was provided for the indicated object. |
| User response: | Collect support data by selecting Help > Collect Support Data and forward it to customer support. | User response: Provide more authentication information to the indicated object and try again. |
| REP01021 | Initialization error occurred "{0}". | REP01031 Repository storage table structure is obsolete. Contact repository storage administrator. |
| Explanation: | An error occurred while the repository was being initialized. | Explanation: An error occurred while trying to access a repository that has an obsolete structure. |
| User response: | Check the repository initialization parameters and try again. | User response: Contact your administrator to upgrade the repository database structure. |
| REP01022 | Invalid initialization parameters were specified "{0}". | REP01032 CCSID information could not be found for current QMF catalog. |
| Explanation: | An error occurred while the repository was being initialized. | Explanation: An error occurred while trying to read CCSID information from the QMF Catalog. |
| User response: | | User response: Specify valid CCSID information and try again. |

REP01034 **A repository with name {0} already exists.**

Explanation:

The repository could not be created, because a repository of the same name already exists.

User response:

Specify a different name for the new repository and try again.

REP01035 **Unknown object type.**

Explanation:

An error occurred while trying to access an unsupported object.

User response:

Ensure that all of the required plug-ins are installed and available.

REP01036 **Object {0} cannot be added to this folder.**

Explanation:

The indicated object can not be added to the specified folder.

User response:

Specify a different folder in which to add the indicated object.

REP01037 **Repository could not be found.**

Explanation:

The specified repository could not be found.

User response:

Check the repository name for errors or specify a different repository name and try again.

REP01038 **Undefined or wrong repository identifier**

Explanation:

An error occurred while trying to initialize a repository with deprecated initialization parameters.

User response:

Change the repository connection properties through the **Edit Repository Connection** dialog and try again.

REP01039 **There are no catalogs on the server**

Explanation:

No QMF catalogs exist on the specified server.

User response:

Specify a different server and try again.

REP01042 **You are not authorized to use objects with Owner = {0}.**

Explanation:

Your user ID is not authorized to use objects with the indicated owner.

User response:

Specify an object that has a different owner or contact your administrator for more information.

REP01043 **This data source cannot be accessed by QMF**

Explanation:

The application can not access this data source.

User response:

Check your user license and try again.

REP01044 **Access to data source "{0}" blocked due to QMF catalog restriction violation.**

Explanation:

Access to the indicated data source is blocked due to a restriction imposed by the QMF catalog protection limits.

User response:

Specify a different data source to access or contact your administrator for more information.

REP01045 **Associated QMF catalog data source could not be found: {0}**

Explanation:

The indicated QMF catalog data source could not be found.

User response:

Check the data source name for errors or specify a different data source and try again.

REP01046 **Cannot find connection: {0}**

Explanation:

The indicated repository connection could not be found.

User response:

Specify a different repository connection and try again.

REP01047 **Cannot find data source: {0}**

Explanation:

The indicated data source could not be found.

User response:

Specify a different data source and try again.

REP01049 **Object name cannot exceed {0} characters.**

Explanation:

The indicated object name exceeds the maximum character limit.

User response:

Shorten the indicated object name and try again.

REP01050 **Object path cannot exceed {0} characters.**

Explanation:

The indicated object path exceeds the character limit.

User response:

Shorten the indicated object path and try again.

REP01051 **Cannot connect to repository "{0}" because it is used by another instance of QMF. Select different repository.**

Explanation:

You can not connect to the indicated repository because it is being used by another instance of the application.

User response:

Select a different repository connection and try again.

REP01052 **Data Source plug-in "{0}" was not properly initialized.**

Explanation:

The indicated plug-in was not properly initialized.

User response:

Change the plug-in parameters.

RGEN1000 **An error has occurred while generating the report: {0}**

Explanation:

The indicated error occurred while the report was being generated.

User response:

Resolve the indicated error and retry to generate the report.

RGEN1001 **The action was cancelled.**

Explanation:

The specified action was cancelled

User response:

Rerun the specified action.

RGEN1002 **The number of columns in the form does not match the number of columns in the query.**

Explanation:

The number of columns in the form does not match the number of columns in its associated query.

User response:

Edit the number of columns in the form to match the number of columns its associated query.

RGEN1003 **Internal error.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RGEN1004 **The number of columns in the form does not match the number of columns in the query.**

Explanation:

The number of columns in the form does not match the number of columns in its associated query.

User response:

Edit the number of columns in the form to match the number of columns its associated query.

RGEN1005 **The data source object is not specified.**

Explanation:

The data source object for the report is not specified.

User response:

Specify a data source object for the report and try again.

RLE00002 **Cannot retrieve resource limits of type {0}.**

Explanation:

The resources of the indicated type can not be retrieved.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RLE00003 **An error has occurred while retrieving resource limits from remote site: {0}**

Explanation:

The indicated error occurred while resource limits were being obtained from a remote site.

User response:

Resolve the indicated error and try again.

RLE00004 **Unknown resource limits key is used to retrieve limits.**

Explanation:

An error occurred while retrieving resource limits.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RLE00005 **An error has occurred while obtaining resource limits: {0}**

Explanation:

The indicated error occurred while resource limits were being obtained.

User response:

Resolve the indicated error and try again.

RLE00006 **Cannot retrieve resource limits because resource limits key content is corrupted. {0}**

Explanation:

The indicated error occurred while attempting to retrieve resources limits.

User response:

Resolve the indicated error and try again.

RLE00007 **The following error has occurred while retrieving resource limits: {0}**

Explanation:

The indicated error occurred while resource limits were being retrieved.

User response:

Resolve the error and try again.

RLE00008 **Data source cannot be accessed.**

Explanation:

Your user ID does not have permission to access the data source.

User response:

Check your resource limits by selecting **Resource Limits** from the **View** menu or contact your administrator for more information.

RLE00009 **LOB data saving is denied.**

Explanation:

Your user ID does not have permission to save objects that contain LOB data.

User response:

Check your LOB Options resource limits by selecting **Resource Limits** from the **View** menu or contact your administrator for more information.

RLE00010 **You are only authorized to run saved queries at this server.**

Explanation:

You are only allowed to run saved queries while you are connected to the current server.

User response:

Check you Save Data resource limits by selecting **Resource Limits** from the **View** menu or contact your administrator for more information.

RLE00011 **You are not authorized to save objects at this server.**

Explanation:

You are not authorized to save objects while you are connected to the current server.

User response:

Check your Save Data resource limits by selecting **Resource Limits** from the **View** menu or contact your administrator for more information.

RLE00012 **You are not authorized to access data source {0}.**

Explanation:

Your user ID is not authorized to access the indicated data source.

User response:

Try to connect to another data source, or contact your administrator for information regarding specific allowances.

RLE00013 **You are not authorized to access data source {0}.**

Explanation:

Your user ID is not authorized to access the indicated data source.

User response:

Try to connect to another data source, or contact your administrator for information regarding specific allowances.

RLW00001 **{0} rows were retrieved. The Maximum Rows to Fetch warning limit has been exceeded.**

Explanation:

The indicated number of rows were retrieved and the warning limit for the **Maximum Rows to Fetch** limit has been reached.

User response:

Click **OK** to continue running the query. Click **Abort** to abort the query. Click **Ignore** to ignore the warning.

RLW00002 **{0} bytes were retrieved. The Maximum Bytes to Fetch warning limit has been exceeded.**

Explanation:

The indicated number of bytes were retrieved and the warning limit for the **Maximum Bytes to Fetch** limits has been reached.

User response:

Click **OK** to continue running the query. Click **Abort** to abort the query. Click **Ignore** to ignore the warning.

RLW00003 **There has been no response from {0}. {1} for {2} seconds.**

Explanation:

The query was cancelled because the Server Response Timeout limit has been reached. There was no response from the indicated server after a wait time of the indicated length.

User response:

Check your Server Response Timeouts resource limit by selecting **Resource Limits** from the **View** menu or try to connect to the server during a time of lighter server traffic.

RLW00004 **The query has been idle for {0} seconds.**

Explanation:

The query has been idle for the indicated number of seconds and the **Idle Query Timeouts** warning limit has been reached.

User response:

Click **OK** to continue running the query. Click **Abort** to abort the query. Click **Ignore** to ignore the warning.

RLW00005 **The query was cancelled because the Idle Query Timeout limit was exceeded.**

Explanation:

The query was cancelled because the Idle Query Timeout limit was exceeded.

User response:

Attempt to rerun the query at a time when there is less server traffic.

RLW00006 **The operation cannot be performed because the processing of LOB data is disabled.**

Explanation:

The operation can not be performed because the LOB Retrieval Option resource limits is disabled.

User response:

Check your LOB Retrieval Option resource limit by selecting **Resource Limits** from the **View** menu or contact your administrator for more information.

RPT00001 **Unable to process a Level component.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT00002 **Report does not contain a main page.**

Explanation:

The visual report does not contain a main page.

User response:

Make sure that the report contains a main page.

RPT00003 **A group expression "{0}" is not correct.**

Explanation:

The indicated expression refers to a nonexistent column.

User response:

Ensure that the expression refers to a valid column.

RPT00014 **BLOB locator was not initialized.**

Explanation:

An error occurred while retrieving a BLOB locator.

User response:

Reconnect to the repository and try again.

RPT00015 **Query results were not initialized.**

Explanation:

The query results were not initialized.

User response:

Reconnect to the repository and try again.

RPT00033 **Internal swing renderer error: too early completion.**

Explanation:

The line object is incorrect.

User response:

Delete the line object and recreate it in the designer.

RPT00034 **Internal swing renderer error: incorrect segment type.**

Explanation:

The line object is incorrect.

User response:

Delete the line object and recreate it in the designer.

RPT00035 **Internal error in reporter engine.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT00037 **Invalid arguments was used: row and column should be positive or zero.**

Explanation:

A row or column in the grid is of a length that is less than zero when viewed in the HTML or SWF rendering mode.

User response:

Reload the visual project.

RPT00038 **Parameter {0} already exists.**

Explanation:

The specified parameter already exists.

User response:

Specify a new name for the new parameter.

RPT00039 **Unable to process one or more expressions in the visual dashboard to render its runtime content.**

Explanation:

The renderer attempted to render a scene that is undefined.

User response:

Reload the visual project.

RPT00040 **A renderer for the selected report format has generated an error. No output contents has been produced.**

Explanation:

A renderer for the selected report format has generated an error. No output contents have been produced.

User response:

Try to view the visual project in another rendering mode, or reload the visual project.

RPT00041 **Internal error: unable to create reporter engine.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT00042 **Unable to parse previously stored visual project data.**

Explanation:

The visual project's data is corrupted.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT00043 **Unable to locate visual dashboard format file.**

Explanation:

{CONFIG_DIR}/converter/repository/world.xml can not be found or read.

User response:

Reinstall QMF.

RPT00044 **Unable to read visual dashboard format file.**

Explanation:

{CONFIG_DIR}/converter/repository/world.xml can not be found or read.

User response:

Reinstall QMF.

RPT00045 **Unable to read visual dashboard scene.**

Explanation:

The visual project's data is corrupted.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT00048 **Unable to serialize object {0}.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT00049 **Internal serialization error: no tag for {0} object class.**

Explanation:

The tag description for the indicated class could not be found during the serialization process.

User response:

Reinstall QMF.

RPT00050 **Unknown parameter type.**

Explanation:

An unknown parameter type was encountered during the serialization process.

User response:

Reload the visual project.

RPT00051 **Invalid action type.**

Explanation:

An unknown action type was encountered while loading the visual project.

User response:

Reload the visual project.

RPT00052 **Embedded visual dashboard picture is corrupted.**

Explanation:

An embedded image in the visual dashboard is corrupted and can not be opened.

User response:

Check all of the embedded images in the visual dashboard and delete any that are corrupted.

RPT00053 **Invalid color value.**

Explanation:

Corrupted tags were encountered while loading a ColorMap or a ColorSequence.

User response:

Reload the visual project.

RPT00054 Invalid threshold value for color map.

Explanation:

Corrupted tags were encountered while loading a ColorMap or a ColorSequence.

User response:

Reload the visual project.

RPT00055 Invalid event type.

Explanation:

An unknown event type was encountered.

User response:

Reload the visual project.

RPT00056 Missing image alias.

Explanation:

An unknown image was found.

User response:

Reload the visual project.

RPT00057 Invalid number value.

Explanation:

While loading the visual project, a string that was supposed to contain a number value was encountered without digit characters.

User response:

Reload the visual project or try using a different locale.

RPT00058 Invalid date value.

Explanation:

While loading the visual project, a string that was supposed to contain a date value was encountered with characters that could not be parsed into a date value.

User response:

Reload the visual project or try using a different locale.

RPT00059 Invalid point value.

Explanation:

While loading the visual project, a string that was supposed to contain a point value was encountered with characters that could not be parsed into a point value.

User response:

Reload the visual project.

RPT00060 Unable to read visual dashboard.

Explanation:

An error occurred while loading the visual dashboard.

User response:

Reload the visual dashboard.

RPT00061 (Multiple possible messages; see Explanation)

Explanation

1. Missing query alias. - A query with no query name tag was encountered while loading the visual project.
2. The offline visual report cannot be saved in file. - Offline visual reports can not be saved to a file.

User response

1. Reload the visual project.
 2. Convert the visual report to an online report before saving to a file.
-

RPT10001 Source Visionary World file has not been specified.

Explanation:

The Visionary World file can not be found.

User response:

Ensure that the Visionary World file exists and that it can be read.

RPT10002 Configuration directory has not been specified.

Explanation:

The configuration directory for the converter can not be found.

User response:

Reinstall QMF.

RPT10003 Source Visionary World file has not been found.

Explanation:

The Visionary World file can not be found.

User response:

Ensure that the Visionary World file exists.

RPT10004 Configuration directory has not been found.

Explanation:

The configuration directory for the converter can not be found.

User response:

Reinstall QMF.

RPT10005 Configuration file {0} has not been found.

Explanation:

The configuration file for the converter can not be found.

User response:

Reinstall QMF.

RPT10006 Unable to locate Visionary Connection file(s) in the folder {0}.

Explanation:

The Visionary Connection files do not exist in the proper locations.

User response:

Check that all Visionary World files are saved in the correct locations.

RPT10007 Unable to read Visionary Query file {0}.

Explanation:

The Visionary World file that is specified in the query can not be found.

User response:

Check that all Visionary World files are saved in the correct locations.

RPT10008 Unable to locate Visionary Scene file {0}.

Explanation:

The indicated scene file is not saved in the correct location.

User response:

Check that all Visionary World files are saved in the correct locations.

RPT10009 Unable to read an image file {0}.

Explanation:

The image file that is specified by the indicated name can not be found.

User response:

Check that all Visionary World files are saved in the correct locations.

RPT10010 Picture file {0} is too large.

Explanation:

The indicated picture file is too long.

User response:

Specify a smaller picture file to be used in the visual project.

RPT10011 XML parsing error.

Explanation:

An XML parsing error occurred while attempting to convert the Visionary World.

User response:

Save the Visionary World file in Visionary and try again.

RPT10012 XML reading error.

Explanation:

An input/output error occurred while attempting to read the XML data.

User response:

Try to convert the XML data again.

RPT10013 Unable to create Visual Dashboard object.

Explanation:

An error occurred while attempting to convert the Visionary World file.

User response:

Try to convert the Visionary World file again.

RPT10014 Cannot parse configuration file {0}.

Explanation:

An error occurred while attempting to parse the converter configuration files.

User response:

Reinstall QMF.

RPT10015 Visual Dashboard for a connection is not specified.

Explanation:

The Visionary World file was not loaded properly.

User response:

Reload the Visionary World file.

RPT10016 Invalid color description has been found.

Explanation:

Corrupted tags were encountered while loading a ColorMap or ColorSequence.

User response:

Reload the Visionary World file.

RPT10017 Invalid color map description has been found.

Explanation:

Corrupted tags were encountered while loading a ColorMap.

User response:

Reload the Visionary World file.

RPT10018 Unknown tag: {0}

Explanation:

Unknown tags were encountered while attempting to convert the Visionary World file.

User response:

Try to convert the Visionary World file again.

RPT10019 Invalid scene template description.

Explanation:

An error occurred while loading the Visionary World scene file.

User response:

Reload the Visionary World scene file.

RPT10020 Corrupted property value has been found.

Explanation:

An incorrect format or a corrupted value was specified in a property tag.

User response:

Reload the Visionary World file.

RPT10021 Invalid URL or path format: "{0}"

Explanation:

The indicated URL or path format is invalid.

User response:

Check the URL for errors or specify a different URL.

RPT10022 Unknown or unsupported event type

Explanation:

A corrupted or unknown event type was encountered.

User response:

Reload the Visionary World file.

RPT10023 Function name in expression is missing

Explanation:

A function name in an expression is missing.

User response:

Reload the Visionary World file.

RPT10024 Cannot parse function argument

Explanation:

The function argument could not be parsed while loading the Visionary World file.

User response:

Reload the Visionary World file.

RPT10025 Invalid scene template description.

Explanation:

An error occurred while loading the Visionary World scene file.

User response:

Reload the Visionary World file.

RPT10026 Unable to parse a number: {0}

Explanation:

While loading the Visionary World file, a string that was supposed to contain a number value was encountered without digit characters.

User response:

Reload the Visionary World file.

RPT10027 PointSet component XML is corrupted.

Explanation:

While loading the Visionary World file, a string that was supposed to contain a point value was encountered with characters that could not be parsed into PointSet value.

User response:

Reload the Visionary World file.

RPT10028 Unknown expression type {0}.

Explanation:

An unknown expression type was encountered while loading the Visionary World file.

User response:

Reload the Visionary World file.

RPT10029 One or more queries do not have alias.

Explanation:

A query alias tag that does not contain a name property was encountered while converting the Visionary World file.

User response:

Reload the Visionary World file.

RPT10031 Unable to parse a point value.

Explanation:

While loading the Visionary World file, a string that was supposed to contain a point value was encountered with characters that could not be parsed into PointSet value.

User response:

Reload the Visionary World file.

RPT10032 Visionary World XML has been corrupted.

Explanation:

An error occurred while attempting to load the Visionary World file.

User response:

Reload the Visionary World file.

RPT10033 Internal error: unable to access Visionary World.

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT10034 Unable to add parameter {0} to a Visual Dashboard

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT20001 Unable to access repository during Visual Report conversion.

Explanation:

The repository could not be accessed while the Visual Report was being converted.

User response:

This could possibly be because you were not connected to the repository. Make sure you are connected and try again.

RPT20002 Source visual report file has not been specified.

Explanation:

The source visual report file has not been specified.

User response:

Specify the source visual report file.

RPT20003 Source visual report file has not been found.

Explanation:

The source visual report file could not be found.

User response:

Specify a valid source visual report file.

RPT20004 Visual Report name has not been specified.

Explanation:

The visual report name has not been specified.

User response:

Specify a name for the visual report.

RPT20005 Unable to create visual report.

Explanation:

An error occurred while creating the visual report.

User response:

Reload the visual report.

RPT20006 Unable to open an embedded image.

Explanation:

An embedded image file was not found.

User response:

Reload the visual report.

RPT20007 Unknown section type: {0}

Explanation:

The structure of a section set and type was not found while loading the visual report.

User response:

Reload the visual report.

RPT20008 Unable to create Visual Report page component.

Explanation:

An error occurred while parsing and creating a page component.

User response:

Reload the visual report.

RPT20009 Internal error: unable to access Visual Report.

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT20011 Unable to set a line style for a line control.

Explanation:

An undefined line style was encountered while loading the visual report.

User response:

Reload the visual report.

RPT20012 Invalid linked picture alias has been found in Visual Report.

Explanation:

A link to a picture without a name was encountered while parsing the visual report data.

User response:

Reload the visual report.

RPT20013 Unable to create Visual Report control.

Explanation:

An error occurred while loading a primitive or control object.

User response:

Reload the visual report.

RPT20015 Invalid parameter name "{0}" has been found.

Explanation:

The parameter value is invalid.

User response:

Specify a different name for the parameter indicated.

RPT20016 Invalid parameter value "{0}" has been found.

Explanation:

The parameter value is invalid.

User response:

Specify a different value for the parameter indicated.

RPT30001 **Navigate to Embedded Scene: Embedded Scene not found for name "{0}".**

Explanation:

The **Navigate to Embedded Scene** action failed because the indicated embedded scene could not be found.

User response:

Correct the name of the embedded scene that is associated with the **Navigate to Embedded Scene** action, or specify another.

RPT30002 **Navigate back: can't execute exit action (maybe history is empty).**

Explanation:

The **Navigate Back** action failed, possibly because the navigation history is empty.

User response:

Reload the visual project and try again.

RPT30021 **DesignReporter: LevelTemplate is null.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30022 **DesignReporter: unknown component {0}.**

Explanation:

An unknown component was encountered during runtime rendering.

User response:

Reload the visual project.

RPT30024 **The template is invalid. FillStyle or LineStyle property is null!**

Explanation:

An error occurred while running the substitution rule to change the Wedge template.

User response:

Check the FillStyle and LineStyle properties for errors.

RPT30025 **The template is invalid. Height == {0}!**

Explanation:

An error occurred while running the substitution rule to change a template.

User response:

Check the Height property for errors.

RPT30026 **The template is invalid. Width == {0}!**

Explanation:

An error occurred while running the substitution rule to change a template.

User response:

Check the Width property for errors.

RPT30028 **Cannot load resources for package {0} and locale {1}.**

Explanation:

The bundle for the resource can not be found.

User response:

Restart the application. If this does not fix the problem, reinstall the application.

RPT30029 **Request for resource string "{0}" from package {1} failed.**

Explanation:

An error occurred while requesting a string resource.

User response:

Restart the application. If this does not fix the problem, reinstall the application.

RPT30035 **No OutputComponentCreator in map for {0}.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30038 **Could not find {0}.**

Explanation:

One of the objects in the **Connections** folder could not be found.

User response:

Reload the project or delete the connection and try again.

RPT30040 **Chart cannot read query result set.**

Explanation:

An error occurred while creating a ClusterGraph, OrganizationChart, or TreeChart.

User response:

Reconnect to the repository and reload the visual project.

RPT30042 **Error - axisLength is infinite. totalPercentages:{0} m_sceneRuntimePanel.getZoomFactor() {1}**

Explanation:

The value of an axis's PercentageLength property is invalid.

User response:

Check the PercentageLength property value of the axis for errors.

RPT30044 **Unknown checkbox value: {0}.**

Explanation:

A value other than UNCHECKED, CHECKED, or INDETERMINATE was specified.

User response:

Ensure that any action that sets the value of a checkbox sets it to UNCHECKED, CHECKED, or INDETERMINATE.

RPT30045 **Unknown m_style: {0}.**

Explanation:

An invalid style property was specified for a ComboBox object.

User response:

Check the style property for the ComboBox object for errors.

RPT30047 **Line is not horizontal or vertical.**

Explanation:

A segment in an ElbowConnector object is set to neither vertical or horizontal.

User response:

Reload the visual project.

RPT30048 **Unsupported segment type: {0}.**

Explanation:

An unsupported segment type was encountered for an ElbowConnector object.

User response:

Reload the visual project.

RPT30049 **Could not get corner type last: {0} start: {1} end: {2}.**

Explanation:

The start and end coordinates of a line segment do not define a corner in an ElbowConnector object.

User response:

Reload the visual project, or delete and recreate the ElbowConnector object.

RPT30050 **Should not have gotten here.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30052 **Unknown horizontal align: {0}.**

Explanation:

A value other than LEFT, RIGHT, or CENTER was encountered in the HorizontalAlign property of an object.

User response:

Ensure that any action that sets the value of the HorizontalAlign property of an object sets it as LEFT, RIGHT, or CENTER.

RPT30053 **Unknown vertical align: {0}.**

Explanation:

A value other than TOP, BOTTOM, or CENTER was encountered in the VerticalAlign property of an object.

User response:

Ensure that any action that sets the value of the VerticalAlign property of an object sets it as TOP, BOTTOM, or CENTER.

RPT30054 **Unknown location anchor: {0}.**

Explanation:

A value other than BOTTOMLEFT, LEFTCENTER, TOPLEFT, BASELINELEFT, BOTTOMCENTER, CENTER, TOPCENTER, BASELINECENTER, BOTTOMRIGHT, RIGHTCENTER, or BASELINERIGHT was encountered in the LocationAnchor property of an object.

User response:

Ensure that any action that sets the value of the LocationAnchor property of an object sets it as BOTTOMLEFT, LEFTCENTER, TOPLEFT, BASELINELEFT, BOTTOMCENTER, CENTER, TOPCENTER, BASELINECENTER, BOTTOMRIGHT, RIGHTCENTER, or BASELINERIGHT.

RPT30055 **Unknown arrowheadStyle: {0}.**

Explanation:

A value other than NONE, TRIANGLE, CIRCLE, DIAMOND, or SWEPTPOINTER was encountered in the MarkerShape property of an arrow object.

User response:

Ensure that any action that sets the value of the MarkerShape property of an arrow object sets it as NONE, TRIANGLE, CIRCLE, DIAMOND, or SWEPTPOINTER.

RPT30056 **Unknown marker shape: {0}.**

Explanation:

A value other than HOLLOWCIRCLE, HOLLOWDIAMOND, HOLLOW SQUARE, HOLLOWTRIANGLE, HOLLOWTRIANGLE2, HOLLOWSTAR, HOLLOWPLUS, SOLIDCIRCLE, SOLIDDIAMOND, SOLIDSQUARE, SOLIDTRIANGLE, SOLIDTRIANGLE2, SOLIDSTAR, SOLIDPLUS, CROSS, PLUS, BOXCROSS, or BOXPLUS was encountered in the MarkerShape property of a data point object.

User response:

Ensure that any action that sets the value of the MarkerShape property of a data point object sets it as HOLLOWCIRCLE, HOLLOWDIAMOND, HOLLOW SQUARE, HOLLOWTRIANGLE, HOLLOWTRIANGLE2, HOLLOWSTAR, HOLLOWPLUS, SOLIDCIRCLE,

SOLIDDIAMOND, SOLIDSQUARE, SOLIDTRIANGLE, SOLIDTRIANGLE2, SOLIDSTAR, SOLIDPLUS, CROSS, PLUS, BOXCROSS, or BOXPLUS.

RPT30060 Unknown attachmentStyle: {0}.

Explanation:

A value other than TOPLEFT, TOPCENTER, TOP, TOPRIGHT, CENTERLEFT, CENTER, CENTERRIGHT, BOTTOMLEFT, BOTTOMCENTER, BOTTOM, or BOTTOMRIGHT was encountered in the AttachmentStyle property of a connector object.

User response:

Ensure that any action that sets the value of the AttachmentStyle property of a connector object sets it as TOPLEFT, TOPCENTER, TOP, TOPRIGHT, CENTERLEFT, CENTER, CENTERRIGHT, BOTTOMLEFT, BOTTOMCENTER, BOTTOM, or BOTTOMRIGHT.

RPT30061 Unknown border type for table: {0}.

Explanation:

An invalid value was encountered in the BorderType property of an object.

User response:

Ensure that any action that sets the BorderType property of an object sets a valid value.

RPT30062 Unknown m_horizontalAlign: {0}.

Explanation:

An invalid value was encountered in the HorizontalAlign property of a table cell.

User response:

Ensure that any action that sets the HorizontalAlign property of a table cell sets a valid value.

RPT30063 Unknown m_verticalAlign: {0}.

Explanation:

An invalid value was encountered in the VerticalAlign property of a table cell.

User response:

Ensure that any action that sets the VerticalAlign property of a table cell sets a valid value.

RPT30064 Unknown border type for table: {0}.

Explanation:

An invalid value was encountered in the Border property of a table.

User response:

Ensure that any action that sets the Border property of a table sets a valid value.

RPT30065 Unknown scrollBarStyle: {0}.

Explanation:

An invalid value was encountered in the ScrollBarStyle property of an object.

User response:

Ensure that any action that sets the ScrollBarStyle property of an object sets a valid value.

RPT30066 Unknown AlignmentStyle: {0}.

Explanation:

An invalid value was encountered in the AlignmentStyle property of a textbox object.

User response:

Ensure that any action that sets the AlignmentStyle property of a textbox object sets a valid value.

RPT30068 Unknown lineStylePattern: {0}.

Explanation:

An invalid value was encountered in the LineStylePattern property of an object.

User response:

Ensure that any action that sets the LineStylePattern property of an object sets a valid value.

RPT30069 Unknown gradient type: {0}.

Explanation:

An invalid value was encountered in the GradientType property of an object.

User response:

Ensure that any action that sets the GradientType property of an object sets a valid value.

RPT30070 Unknown borderType: {0}.

Explanation:

An invalid value was encountered in the BorderType property of a textbox or embedded scene object.

User response:

Ensure that any action that sets the BorderType property of a textbox or embedded scene object sets a valid value.

RPT30071 Unknown imageAnchor: {0}.

Explanation:

An invalid value was encountered in the ImageAnchor property of an object.

User response:

Ensure that any action that sets the ImageAnchor property of an object sets a valid value.

RPT30072 Embedded Scene: Unknown scrollBarStyle.

Explanation:

An invalid value was encountered in the ScrollBarStyle property of an embedded scene object.

User response:

Ensure that any action that sets the ScrollBarStyle property of an embedded scene object sets a valid value.

RPT30073 **Embedded Scene: unknown typeOfFind: {0}.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30076 **The cell chosen by client not found.**

Explanation:

The action can not be performed when using the HTML rendering mode.

User response:

Reload the visual project.

RPT30077 **Grid cell's click event could not be dispatched.**

Explanation:

The action can not be performed when using the HTML rendering mode.

User response:

Reload the visual project.

RPT30079 **Unknown page type - {0}.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer service.

RPT30082 **Parameters group cloning problem {0}.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30083 **No class in map for {0}.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30088 **Unable to serialize an object: {0}.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30089 **Value must be positive: {0}.**

Explanation:

The indicated value must be a positive number.

User response:

Ensure that the indicated value is a positive number.

RPT30090 **Level was null when zoom was {0}.**

Explanation:

No level is specified for the indicated zoom percentage.

User response:

Ensure that a level is specified for the indicated zoom percentage.

RPT30091 **Chart can't find parent column: {0}.**

Explanation:

The indicated parent column could not be found in the result set.

User response:

Ensure that a valid column named is specified in the DataLocator property of the group.

RPT30092 **Chart can't find child column: {0}.**

Explanation:

The indicated child column could not be found in the result set.

User response:

Ensure that a valid column named is specified in the DataLocator property of the group.

RPT30095 **Hierarchy chart must have one DataTemplate child.**

Explanation:

A hierarchy chart must have one DataTemplate child.

User response:

Check the **Project Explorer** and ensure that the hierarchy chart only has one DataTemplate child.

RPT30096 **CurrentRowIndex < 0.**

Explanation:

The specified row index is less than zero. The row index must be an integer that is equal to or greater than zero.

User response:

Ensure the specified row index is an integer that is equal to or greater than zero.

RPT30097 **Unknown interval: {0}.**

Explanation:

An invalid value was specified for the DateAxis property of an object.

User response:

Ensure that any action that sets the value of the DateAxis property of an object sets a valid value.

RPT30098 Unhandled interval: {0}.

Explanation:

An invalid value was specified for the DateAxis property of an object.

User response:

Ensure that any action that sets the value of the DateAxis property of an object sets a valid value.

RPT30099 Unknown dateInterval: {0}.

Explanation:

An invalid value was specified for the DateAxis property of an object.

User response:

Ensure that any action that sets the value of the DateAxis property of an object sets a valid value.

RPT30104 Shouldn't get here - roundIncrement: {0} power: {1}.

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30105 Group name is undefined for group header.

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30106 Group name is undefined for group footer.

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30107 m_templateRows contained a {0}.

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30109 Embedded Scene: Linked not found for name {0}.

Explanation:

The value of the LinkedScene property of an embedded scene is invalid.

User response:

Ensure that the scene that the embedded scene is linked to exists and that the link text is correct.

RPT30111 Unable to clone component {0}.

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30112 Property {0} of component {1} doesn't appear to be of type {2}.

Explanation:

The property of the indicated object is of an invalid type.

User response:

Ensure that any action or expression that sets the value of the property sets it to a valid type.

RPT30114 Swing output cannot be created for component {0}.

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30117 Component {0} does not support action {1}.

Explanation:

The indicated component does not support the indicated action.

User response:

Specify a different action for the component and try again.

RPT30118 Component {0} does not support property {1}.

Explanation:

The indicated component does not support the indicated property.

User response:

Specify a different property and try again.

RPT30120 (Multiple possible messages; see Explanation)

Explanation

1. The global parameter "{0}" does not exist - The indicated global parameter does not exist.

- Query "{0}" does not contain column named "{1}"
 - The indicated column does not exist in the indicated query.

User response

- Check the global parameter name for errors or specify a different global parameter.
- Check the column name for errors or specify a different column name.

RPT30121 Result set index "{0}" for query "{1}" is out of bounds.

Explanation:
An invalid value was set for the ResultSetIndex of a layout object.

User response:
Ensure that any action that sets the ResultSetIndex property of a layout object sets a valid value.

RPT30122 Query with key "{0}" does not exist.

Explanation:
A query with the indicated key does not exist in the repository.

User response:
Select the query in the **Repository Explorer** or **Workspaces** view and copy and paste the key value from the **Properties** view, or specify a different query's key.

RPT30123 Component "{0}" cannot be created due to the following problem: {1}

Explanation:
The indicated component can not be created due to the indicated error.

User response:
Resolve the indicated error and try again.

RPT30124 Result set index cannot be defined.

Explanation:
An internal error occurred.

User response:
Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RPT30126 Security list "{0}" cannot be found.

Explanation:
The indicated security list can not be found.

User response:
Check the security list name for errors or specify a different security list.

RPT30127 Visual project "{0}" cannot be loaded because of missing project descriptor.

Explanation:
The indicated visual project can not be loaded because it is missing a project descriptor.

User response:
Ensure that the visual project has a project descriptor and retry to load the project.

RSE01000 {0}

Explanation:
The indicated error occurred while accessing the repository security provider.

User response:
Resolve the indicated error and try again.

RSE04000 System internal error occurred, please check the error log.

Explanation:
A system internal error occurred.

User response:
Check the error log by selecting **Output** from the context menu of the **Window** menu.

RSE04001 Cannot find User {0}

Explanation:
The indicated user can not be found.

User response:
Check the name of the user for errors and try again.

RSE04002 User {0} already exists;

Explanation:
The indicated user already exists.

User response:
Specify a different user.

RSE04003 Group {0} already exists;

Explanation:
The indicated group already exists.

User response:
Specify a different group.

RSE04004 Cannot find group {0}

Explanation:
The indicated group can not be found.

User response:
Specify a different group.

RSE04005 Not logged in with security provider.

Explanation:

You attempted to access the repository without logging in.

User response:

Login to the repository and try again.

RSE04006 The user ID and password are invalid

Explanation:

The user ID and password that were entered are invalid.

User response:

Check the user ID and password for errors and try again.

RSE04007 Name "{0}" is in conflict with the existing name.

Explanation:

The new name that was entered already exists.

User response:

Select a different name.

RSE04008 Object access denied

Explanation:

Access to the specified object is denied.

User response:

Select a different object to access or contact your administrator for more information.

RSE04011 Cannot remove User {0} from Group {1}

Explanation:

The indicated user can not be removed from the indicated group.

User response:

Select another user to be removed from the group.

RSE04012 Cannot delete User {0}

Explanation:

The indicated user can not be deleted.

User response:

Select another user to be deleted.

RSE04013 Cannot delete Group {0}

Explanation:

The indicated group can not be deleted.

User response:

Select another group to be deleted.

RSE04014 Cannot initialize security provider, {0}

Explanation:

The indicated error occurred while initializing the repository.

User response:

Resolve the indicated error and try again.

RSE04015 This security provider is a read-only one.

Explanation:

You can not change the principals of a read-only security provider.

User response:

Choose another security provider to modify.

RSE04016 Cannot find any name associates with UID {0} in Repository

Explanation:

The indicated repository identifier is invalid.

User response:

Specify a valid repository identifier and try again.

RSE04017 AuthType is not anonymous("none"), please specify principal and credentials

Explanation:

An error occurred while initializing the LDAP security provider.

User response:

Change the repository initialization parameters and try again.

RSE04018 LDAP Configuration contains illegal values or some required values are missing, please check the preferences({0})

Explanation:

An error occurred while initializing the LDAP security provider.

User response:

Change the repository initialization parameters and try again.

RSE04019 Cannot connect to the LDAP server. Communication broken or Directory Service is not available.

Explanation:

An error occurred while connecting to the LDAP directory.

User response:

Check the network connection and try again.

RSE04020 Cannot read User's password attribute (Make sure you have enough permission or PasswordAttribute in configuration is correct.)

Explanation:

An error occurred while reading data from the LDAP directory.

User response:

Ensure that the LDAP service is accessible.

RSE04021 Invalid search filter in LDAP configuration
Explanation:

The specified search filter in the LDAP configuration is invalid.

User response:

Specify a valid search filter for the LDAP configuration.

RSE04022 LDAP server returned error
Explanation:

An error occurred while reading data from the LDAP directory.

User response:

Ensure that the LDAP service is accessible.

RSE04023 Cannot find the specified name {0}
Explanation:

The indicated user name can not be found.

User response:

Specify a different user name.

RSE04024 Not enough permission to perform the LDAP operations.
Explanation:

Your user ID does not have permission to perform LDAP operations.

User response:

Contact your administrator for more information concerning specific permissions and allowances.

RSE04025 Cannot find a attribute of entry {0} specified in LDAP configurations.
Explanation:

An error occurred while reading attribute data from the LDAP configuration.

User response:

Correct the configuration parameters and try again.

RSE04026 UUID {0} mapping to an invalid distinguished name {1}.({2})
Explanation:

The indicated distinguished name is invalid.

User response:

Specify a valid distinguished name and try again.

RSE04028 Incorrect operation.
Explanation:

An error occurred while editing the principals of the security provider.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

RSE04029 Cannot register security provider.
Explanation:

An error occurred while registering the security provider in the provider registry.

User response:

Ensure that the registered name of the provider is unique.

RSE04030 Security provider not initialized.
Explanation:

An error occurred while accessing the security provider service because the service is not initialized.

User response:

Initialize the security provider and try again.

RSE04031 Security provider not registered.
Explanation:

An error occurred while accessing the security provider because the security provider is not registered.

User response:

Register the security provider and try again.

RSE04032 You can not remove default Administrator.
Explanation:

The default administrator can not be deleted.

User response:

Specify a different administrator to be deleted.

RSE04033 You can not change the default Administrator's membership.
Explanation:

The membership of the default administrator can not be changed.

User response:

Specify the membership of a different administrator to change.

RSE04035 You can not rename the Administrators group.
Explanation:

The group of the default administrator can not be renamed.

User response:

Specify the group of a different administrator to be renamed.

RSE04036 Password has expired and is no longer valid.
Explanation:

The specified password has expired and is no longer valid.

User response:

Specify an updated password and try again.

RSE04037 **Group could not become its own parent.**

Explanation:

A group may not be its own parent. You must select a different group as the parent.

User response:

Select a different group to be the parent of the specified group.

SPL01001 **Cannot find objectClass attribute of {0}**

Explanation:

The indicated object does not have an objectClass attribute. Every object in the LDAP directory must contain this attribute.

User response:

Specify an objectClass attribute for the indicated object.

SPL01002 **The returned entry {0} is not acceptable for LDAP Security Provider, its objectClass is not {1} or {2}**

Explanation:

The indicated object is defined as neither a user or group in the LDAP security provider.

User response:

Edit the objectClass attribute of the indicated object to specify whether it is a user or group.

SPL01003 **Cannot read the Group's name (attribute: {0})**

Explanation:

The LDAP directory does not contain an object that has the required attribute.

User response:

Edit the groupName attribute so that it will point to an existing object in the LDAP directory.

SPL01004 **Cannot read the User's name (attribute: {0})**

Explanation:

The LDAP directory does not contain an object that has the required attribute.

User response:

Edit the userName attribute so that it will point to an existing object in the LDAP directory.

SPL01005 **LDAP server host or port was not specified.**

Explanation:

The LDAP server host or port was not specified.

User response:

Specify the LDAP server and host.

SPL01006 **The base distinguished name of the LDAP server must be present.**

Explanation:

The base distinguished name of the LDAP server was not specified.

User response:

Specify the base distinguished name of the LDAP server.

SPL01007 **The distinguish name of user principal must be present.**

Explanation:

The distinguished name of the principal user was not specified.

User response:

Specify the distinguished name of the principal user or use anonymous authentication.

SPL01008 **The credentials of user entry must be specified.**

Explanation:

The credentials of the entry user were not specified.

User response:

Specify the credentials of the entry user or use anonymous authentication.

SPL01009 **Cannot create LDAP entry for administrator user: {0}**

Explanation:

The user who is associated with the administrative account could not be found in the LDAP directory.

User response:

Specify the administrative user of the LDAP directory.

SPL01010 **Cannot create LDAP entry for administrators group: {0}**

Explanation:

The group that is associated with the administrative account could not be found in the LDAP directory.

User response:

Specify the administrative group of the LDAP directory.

SPL01011 **Cannot create LDAP entry for everyone group: {0}**

Explanation:

The group that is associated with the everyone account could not be found in the LDAP directory.

User response:

Specify the everyone group of the LDAP directory.

SPL01012 Cannot create LDAP entry for root group: {0}

Explanation:

The object that is associated with the Base DN could not be found in the LDAP directory.

User response:

Specify the Base DN object in the LDAP directory.

SPL01013 Cannot find the JNDI extension package or Sun's LDAP Service Provider.

Explanation:

The JNDI extension package or Sun's LDAP Service Provider was not located in the CLASSPATH.

User response:

Ensure that the CLASSPATH includes both the JNDI extension package and Sun's LDAP Service Provider.

SPL01014 The catalog object is missing.

Explanation:

The catalog object is not set.

User response:

Specify the catalog object and try again.

SPL01015 Empty name is not allowed.

Explanation:

No name was specified as an RDN.

User response:

Specify a name as an RDN.

SPL01016 Illegal RDN of security object (offending string: "{0}")

Explanation:

The RDN is invalid. The RDN must be one or more LDAP components that are separated by commas.

User response:

Specify valid components for the RDN.

SPL01017 Unknown attribute name "{0}" in RDN.

Explanation:

An unknown attribute name was encountered while parsing the RDN.

User response:

Check the attributes of the RDN for errors.

SPL01101 Currently Client Authentication is not supported.

Explanation:

Client Authentication is not supported.

User response:

Use anonymous authentication.

SPL01103 SSL is not correctly initialized.

Explanation:

The secure socket protocol factory is not initialized.

User response:

Restart the application and try again.

SPL01104 Attribute {0} is absent.

Explanation:

The indicated attribute was not found while reading the LDAP object.

User response:

Specify a valid value for the indicated attribute.

SPL01105 Object class name "{0}" is unknown.

Explanation:

The object name has an invalid class name; it must be either qmfUser or qmfGroup.

User response:

Specify either qmfUser or qmfGroup for the indicated object name.

SQM02001 An input/output error occurred while processing package "{0}". {1}

Explanation:

The indicated error occurred while processing the indicated package.

User response:

Resolve the indicated error and try again.

SQM02002 Error in SQLJ customization: {0}

Explanation:

The indicated error occurred while processing the SQLJ customization.

User response:

Resolve the indicated error and try again.

SQM02003 No SQLJ installed

Explanation:

The SQLJ library could not be loaded.

User response:

Add sqlj.zip to the driver definition.

SQM02004 An error occurred while retrieving profile data for package "{0}". {1}

Explanation:

The indicated error occurred while the profile data for the indicated package was being loaded.

User response:

Resolve the indicated error and try again.

SQM02005 An error occurred while loading the profile for package "{0}".

Explanation:

The indicated error occurred while the profile for the indicated package was being loaded.

User response:

Resolve the indicated error and try again.

SQM02006 **While loading the class with the implementation of package "{0}" the following error has occurred: "{1}".**

Explanation:

The indicated error occurred while class with the implementation of the indicated package was being loaded.

User response:

Reinstall the application and try again.

SQM02007 **A database access error occurred while processing package "{0}". {1}**

Explanation:

The indicated database access error occurred while the package was being processed.

User response:

Resolve the indicated database access error and try again.

SQM02008 **A database access error occurred: {1}**

Explanation:

The indicated database access error occurred.

User response:

Resolve the indicated error and try again.

SQM02009 **User has no rights to execute queries of package "{0}". Original error follows: {1}**

Explanation:

The user does not have permission to access the package because of the indicated reason.

User response:

Contact your database administrator.

SQM02010 **An error occurred while deleting profile for package "{0}". {1}**

Explanation:

The indicated error occurred while the profile for the indicated package was being deleted.

User response:

Resolve the indicated error and try again.

SQM02011 **Empty customizer call pattern**

Explanation:

Package customization failed because of an unexpected error.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

SQM02013 **An error occurred while loading data for package "{0}".**

Explanation:

An error occurred while the data for the indicated package was being loaded.

User response:

Resolve the indicated error and try again.

SQM02014 **Wrong customizer call pattern: {0}**

Explanation:

Package customization failed because of an unexpected error.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

SQM02016 **Cannot create collection "{0}" due to the following: {1}**

Explanation:

The indicated collection could not be created due to the indicated error.

User response:

Resolve the indicated error and try again.

SQM02017 **Profile for package "{0}" is too long to be stored at database**

Explanation:

The profile for the indicated package is too long to be stored at the database.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

SQM02018 **No collection ID (profile owner) is specified. Cannot proceed in static mode.**

Explanation:

The SQLJ package can not be loaded if no collection ID is specified.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

SQM02019 **Cannot prepare statement "{0}" of package "{1}" due to the following error: "{2}"**

Explanation:

The indicated statement of the package can not be prepared due to the indicated error.

User response:

Resolve the indicated error and try again.

SQM02020 **While checking user rights on package "{0}" the following error has occurred: "{1}"**

Explanation:

The indicated error occurred while the user rights on the indicated package were checked.

User response:

Resolve the indicated error and try again.

SQM02021 **Inconsistent profile data for queries package "{0}" were detected. Rebind package.**

Explanation:

Inconsistent profile data for the indicated queries package was detected.

User response:

Rebind the package.

SQM02100 **The required parameter "{0}" is missing.**

Explanation:

The indicated required parameter is missing.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

SQM02101 **An error occurred while initializing the connection. {0}**

Explanation:

The indicated error occurred while the connection was initialized.

User response:

Resolve the indicated error and try again.

SQM02102 **{0}**

Explanation:

The indicated database access error occurred.

User response:

Resolve the indicated error and try again.

SQM02103 **An error occurred while loading package description class "{0}". {1}**

Explanation:

The indicated error occurred while the package description class was loaded.

User response:

Resolve the indicated error and try again.

SQM02104 **An error occurred while loading the profile for package "{0}". {1}**

Explanation:

The indicated error occurred while the profile for the indicated package was loaded.

User response:

Resolve the indicated error and try again.

SQM02105 **Static mode profile for package "{0}" is not found.**

Explanation:

The static mode for the indicated package was not found.

User response:

Reinstall the application and try again.

SQM02106 **An error occurred while creating directory "{0}".**

Explanation:

The temporary directory for the customization process could not be created.

User response:

Ensure that the application has permissions to create the directory and try again.

SQM02107 **An error occurred while creating profiles directory "{0}".**

Explanation:

An error occurred while the indicated profile's directory was created.

User response:

Ensure that the application has permissions to create the directory and try again.

SQM02108 **An error occurred while saving the profile for package "{0}". {1}**

Explanation:

The indicated error occurred while the profile for the indicated package was saved.

User response:

Resolve the error and try again.

SQM02109 **An error occurred while saving the packages description file. {0}**

Explanation:

The indicated error occurred while the package description file was saved.

User response:

Resolve the error and try again.

SQM02110 **An error occurred while starting customization. {0}**

Explanation:

The indicated error occurred while customization was started.

User response:

Resolve the error and try again.

SQM02111 **An error occurred while customizing package "{0}". {1}**

Explanation:

The indicated error occurred while the indicated package was customized.

User response:

Resolve the indicated error and try again.

SQM02112 **An error occurred while creating the profiles data table. {0}**

Explanation:

The indicated error occurred while the data table of the profile was created.

User response:

Resolve the indicated error and try again.

SQM02113 **No user ID list for GRANT/REVOKE operation was specified.**

Explanation:

No user ID list for GRANT/REVOKE operation was specified.

User response:

Specify a user ID list for the GRANT/REVOKE operation.

SQM02115 **An error occurred while uninstalling the packages. {0}**

Explanation:

The indicated error occurred while the packages were uninstalled.

User response:

Resolve the indicated error and try again.

SQM02116 **File with packages description was not found. {0}**

Explanation:

The indicated error occurred while preparing the packages descriptions based on the previously build packages description file.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

SQM02117 **An error occurred while reading the packages description file. {0}**

Explanation:

The indicated error occurred while the packages description file was being read.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

SQM02119 **An error occurred while installing the package {0}. {1}**

Explanation:

The indicated error occurred when the indicated package was being installed.

User response:

Resolve the indicated error and try again.

SQM02120 **An error occurred while uninstalling the package {0}. {1}**

Explanation:

The indicated error occurred when the indicated package was being uninstalled.

User response:

Resolve the indicated error and try again.

SYNERRCD000 **DSS header length less than 6. 1**

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD000 **DSS header length does not match the number of bytes of data found. 2**

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD000 **DSSheader C-byte not D0. 3**

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD000 **DSS header f-bytes either not recognized or not supported. 4**

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD000 **DSS continuation specified but not found. 5**

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD000 **DSS chaining specified but no DSS found. 6**

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD000 **Object length less than four. 7**

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD000 **Object length does not match the number of bytes of data found.**
8

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD000 **Object length greater than maximum allowed.**
9

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERCRD001 **Object length less than minimum required.**
0

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD001 **Object length not allowed.**
1

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD001 **Incorrect large object extended length field.**
2

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD001 **Object codepoint index not supported.**
3

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD001 **Required object not found.**
4

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD001 **Too many command data objects sent.**
5

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD001 **Mutually-exclusive objects present.**
6

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD001 **Too few command data objects sent.**
7

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD001 **Duplicate object present.**
8

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD001 **Invalid request correlator specified.**
9

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD002 **Required value not found.**
0

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD002 **Reserved value not allowed.**
1

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD002 **DSS continuation less than or equal to two.**
2

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD002 **Objects not in required order.**
3

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD002 **DSS chaining bit not 1, but**
4 **DSSFMT bit 3 set to 1.**

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD002 **Previous DSS indicated current**
5 **DSS has the same request**
 correlator, but the request
 correlators are not the same.

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD002 **Previous DSS indicated current**
6 **DSS has the same request**
 correlator, but the request
 correlators are not the same.

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD002 **DSS chaining bit not 1, but error**
7 **continuation requested.**

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD002 **Mutually-exclusive parameter**
8 **values specified.**

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD002 **Codepoint not valid command.**
9

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

SYNERRCD003 **The atmind instance variable is**
0 **not set to its default value of**
 0x00 on an EXCSQLSTT command
 within an atomic EXCSQLSTT
 chain.

Explanation:

An internal error in the driver code occurred.

User response:

Send an error report to the developers.

THM1001 **Cannot copy CSS file.**

Explanation:

The CSS file could not be copied to the server.

User response:

Ensure that the CSS file name is correct and try again.

THM1002 **Cannot extract main theme CSS**
 file from the archive.

Explanation:

The specified archive does not contain the main CSS file.

User response:

Ensure that the main CSS file is saved to the archive and try again.

THM1003 **Already registered. Try another**
 theme name.

Explanation:

A theme with the same name already exists.

User response:

Specify a different name for the theme.

THM1004 **Theme root folder already exists.**
 Try another theme name.

Explanation:

A theme with the same name already exists.

User response:

Specify a different name for the theme.

UIM12001 **Internal error has occurred: {0}**

Explanation:

The indicated internal error occurred.

User response:

Resolve the indicated error and try again.

UIM12004 **An input/output error occurred:**
 {0}

Explanation:

The indicated input/output error occurred.

User response:

Resolve the indicated error and try again.

UIM12005 **The file does not exist: {0}**

Explanation:

The indicated file does not exist.

User response:

Specify a valid file and try again.

VAL00001 **Variable {0} not found.**

Explanation:

The indicated variable was not found.

User response:

Specify a valid variable.

VAL00003 **Type conversion error. Object of {0} type cannot be converted to {1} type.**

Explanation:

The indicated object can not be converted to the indicated type.

User response:

Specify an object of a different type to be converted, or convert the object to a different type.

VAL00004 **Unit conversion error. Unit {0} cannot be converted to {1}.**

Explanation:

The indicated unit can not be converted to the indicated type.

User response:

Specify a different unit to be converted, or convert the unit to a different type.

VAL00005 **An expression contains a syntax error.**

Explanation:

An expression that you entered contains a syntax error.

User response:

Check the expressions that you typed for syntax errors.

VAL00006 **Wrong expression has been used.**

Explanation:

An incorrect expression has been used.

User response:

Specify a valid expression.

VAL00007 **Name "{0}" has invalid character at {1}. The name must start with a letter and should not contain special characters. See the help for a list of invalid characters.**

Explanation:

The indicated name contains an invalid character at the indicated column.

User response:

Check the help for syntax rules and retype the invalid character.

VAL00008 **Name cannot be empty. The name must start with a letter and should not contain special characters. See the help for a list of invalid characters.**

Explanation:

The name of a control can not be empty.

User response:

Specify a correct name for the control.

VAL00009 **Unable to process URL: "{0}"**

Explanation:

The indicated URL could not be processed.

User response:

Check the URL for errors and try again.

VAL00010 **Unable to process URI: "{0}"**

Explanation:

The indicated URI could not be processed.

User response:

Check the URI for errors and try again.

VAL00019 **Cycle has been found in definition of following properties: {0}.**

Explanation:

The expression contains a cyclic reference.

User response:

Edit the expression to remove the cyclic reference and try again.

VAL00021 **Cannot evaluate properties: {0}.**

Explanation:

The expression could not be evaluated.

User response:

Check the expression for syntax errors and try again.

VAL00022 **Validation failed: Cannot set {0} to {1}. Value must be {2} {3}.**

Explanation:

The property can not have the indicated value. Possible values are listed in the error message.

User response:

Edit the property's expression and try again.

VDS1000 **{0}**

Explanation:

The indicated error occurred while processing the virtual data source.

User response:

Resolve the indicated error and try again.

VDS1001 **Internal Error.**

Explanation:

An internal error occurred.

User response:

Collect support data by selecting **Help > Collect Support Data** and forward it to customer support.

VDS1002 **No data source was specified for the query.**

Explanation:

No data source was specified for the query.

User response:

Specify a data source for the query.

VDS1003 **Configuration directory is not defined.**

Explanation:

The configuration directory is not defined.

User response:

Define the configuration directory.

VDS1004 **Source data source not found for virtual table "{0}".**

Explanation:

The source data source for the indicated virtual table was not found. It is possible that the source data source has been moved or deleted.

User response:

Check the location of the source data source and copy the source table to the virtual data source again if necessary.

VDS1005 **Table is not defined in the query.**

Explanation:

The table is not defined in the query.

User response:

Define the table in the query.

VDS1006 **Source table column not found for virtual column "{0}" in the table "{1}".**

Explanation:

The indicated virtual column does not match any of the source table columns.

User response:

Ensure that every column in the virtual table has a corresponding column in the source table.

VDS1007 **Only tables and queries can be added to the virtual data source.**

Explanation:

You attempted to move an object that was not a table or a query to the virtual data source. Only tables and queries may be added to the virtual data source.

User response:

Specify a table or query to be moved to the virtual data source.

VDS1008 **Cannot find specified tables in data source.**

Explanation:

The tables that are referenced in the virtual data source can not be found in the source data source. This may be because the source tables were moved or deleted.

User response:

Check the location of the source tables in the data source and copy them to the virtual data source again if necessary.

VDS1009 **Cannot remove the last column from virtual table.**

Explanation:

You attempted to remove the last column from the virtual table. A virtual table must have at least one column.

User response:

Ensure that the virtual table has at least one column.

VDS1010 **Only queries for relational data sources can be added.**

Explanation:

You attempted to add a query from a multidimensional data source. Only queries for relational data sources can be added to virtual data sources.

User response:

Specify queries for relational data sources to be added to the virtual data source.

VDS1011 **Source table not found "{0}".**

Explanation:

The indicated source table was not found. This may be because the source table was moved or deleted.

User response:

Check the location of the source table and copy it to the virtual data source again if necessary.

VDS1012 **Source query not found "{0}".**

Explanation:

The indicated source query was not found. This may be because the source query was moved or deleted.

User response:

Check the location of the source query and copy it to the virtual data source again if necessary.

VDS1013 **Only SQL queries with SELECT statement can be used to create virtual tables.**

Explanation:

You attempted to create a virtual table using a source query that does not contain a SELECT statement. Only queries with SELECT statements can be used as source queries.

User response:

Specify a query that contains a SELECT statement as a source query.

VDS1014 **Virtual table cannot be created because source query returns columns with ambiguous names.**

Explanation:

The virtual table sources a query that returns a result set that has columns with identical names.

User response:

Change the result set column names in the source query and try again.

VDS1015 **Virtual table cannot be created because source query returns one or more unnamed columns.**

Explanation:

The source table of a virtual table must not return unnamed columns.

User response:

Specify a source query that does not return unnamed columns.

VDS1016 **Virtual table cannot be created because source query returns multiple result sets.**

Explanation:

Virtual tables can only be created using source tables that return single result sets.

User response:

Specify a source table that returns a single result set.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol ([®] or [™]), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at <http://www.ibm.com/privacy> and IBM’s Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at <http://www.ibm.com/software/info/product-privacy>.

Glossary

The glossary provides brief descriptions of product terms.

accessibility

Features that help those with physical disabilities, such as restricted mobility or limited vision, use their computer.

breakpoints

Color maps that specify the value at which the color changes.

calculated columns

Columns of data that you add to the query results.

classic reports

Text-based, tabular reports that are generated using query results as the data source and a form template.

color map

A color map can be used to vary the color of a visual project object depending on a value associated with the object.

color sequence

A color sequence can be used to modify the color properties of a visual project object based on an index value.

command bar

An interface that allows you to run procedure commands directly from the QMF for Workstation interface.

connectivity tool

An interface that allows you to define dependencies between non-query driven objects such as text box objects and label objects. The Connectivity tool is only available when creating visual dashboards.

cube structure tree

In the layout format, the Cube Structure tree lists the dimensions and measures that are contained in the cube.

data sources

Data sources allow you to create virtual tables that can be edited and manipulated by users without altering the original tables' content. Virtual data sources work by introducing a metadata layer which mediates between an administrator-defined, virtual data source and the underlying data sources that contain the physical tables and views.

data templates

Data templates are used to create and control the visual representation of query result set data within layout objects.

Diagram Query Editor

When building queries using the Diagram Query editor, you supply table, join, column, sort, and row information and the Diagram Query editor constructs the Structured Query Language (SQL) statements. The Diagram Query editor is not available in QMF for WebSphere.

Draw Query Editor

Use the Draw Query editor to create a query object automatically based on the query type and tables that you specify.

drilldown

A navigation feature that enables your users to click an object and jump to another scene that contains more information that relates specifically to the original object's value. The **Drilldown** wizard is available for all layout objects that can display query result data.

edit codes

Edit codes specify how to format character, date, graphic, numeric, time and timestamp data that will appear in a specific column of a report. You can also create user-defined edit codes.

embedded scene

An embedded scene object is a container through which you can view all or a portion of another scene and then navigate to that scene. The scene where you place the embedded scene object is considered the source scene. The destination scene is the embedded scene.

event actions

You define event actions to objects to provide scene interactivity and navigation. As a user navigates your dashboard, mouse actions trigger predefined events, such as jumps between scenes, executing of SQL statements, or the launching of other applications.

Expression Designer

The Expression Designer is used to specify any property value that is associated with a selected object. The Expression Designer is an alternative way of specifying an object's properties. For each property that can be specified for an object, you are given templates that can help you format the value of that property. In addition, you have more space available to work and view your property value (which can grow quite large if expressions are used).

fast save

When saving your query results data using a "fast save" method, all the processing occurs at the database. There is no further passing of data between the QMF for Workstation/WebSphere interface and the database. The QMF for Workstation/WebSphere interface adds SQL to the original query, the query is rerun and the data is saved directly into the specified table(s). Saving query results data using the "fast save" process significantly improves performance when a large amount of data is to be saved.

file-based repositories

If you are using QMF for Workstation, you can create a file repository connection to set up the information that enables you to connect to a file-based repository that is stored on a local or network drive. File connections can only be created for legacy, file-based repositories.

formatting options

You can customize how the query results will appear in the editor window. You can specify different fonts, colors and text alignments for each column heading and each cell of the column. You can also specify that column and cell formatting be applied based on the results of a conditional expression. You can apply formatting options to entire columns, individual cells, column headings, and summary cells.

forms

Forms are considered objects and they can be saved in your repository, in the QMF catalog, or in a file. When you open a form object that has been saved, you are actually running the form object to generate the report. When opened, forms automatically use the currently active query results as the data source.

form variables

Form variables are codes that you can insert into text fields to produce information on the report. For example, you can insert a date variable to produce the current date whenever the report is printed.

global parameter

A global parameter is available to your entire visual project and to the user of your visual dashboard. You can pass global parameters when you specify object properties or when you create event actions. You also have the option of making global parameters public at runtime so they are accessible to the user.

global resources

Global resources can be used by all the pages of a visual report or all the scenes of a visual dashboard. You can use global resources for computing object properties.

global variables

Global variables are variables that stay active while the current session of QMF for Workstation/ WebSphere is active. This is in contrast to substitution variables that are active only during the execution of an object (query, procedure, form). For objects that use global variables, the value currently defined for the global variable is used.

grouping and aggregation

Grouping and aggregation options can be applied to query result columns to organize the result data into logical or summarized groupings. By adding grouping and aggregation you can automatically obtain summary information about your data and display the data more logically. For example you can roll up data in a report by department or average departmental commissions.

information zooming

Information zooming is a navigation feature that presents a more detailed view of data as your user zooms in on a scene or a data point in a layout. Every layout object and scene has a default zoom level (100% zoom). You can set up information zooming navigation for whole scenes and for individual layout objects. Information zooming on the layout object allows you to provide more query information on each data point as the user zooms in on a layout object. You use the same query results data for each zoom

level. Information zooming on the scene level allows you to display an entirely different layout and query results data for each zoom level.

JDBC drivers

QMF for Workstation/WebSphere uses JDBC to connect to all database repositories and data sources. QMF for Workstation/WebSphere does not include JDBC drivers. The location of the JDBC drivers that QMF for Workstation/WebSphere will use to connect to repositories and data sources must be defined. QMF for Workstation/WebSphere administrators are responsible for identifying where each instance of the application interface will find the JDBC drivers.

job definition files

Job definition files (.jdfx) are legacy QMF files that can be imported to set up scheduled jobs.

task scheduler

QMF for Workstation/WebSphere has the ability to schedule procedures, queries, quick reports, and visual reports to be executed on a periodic, recurring basis or at a specific date and time. You can schedule your object to run at any given time using your local scheduler or the QMF for WebSphere scheduler.

layout objects

In a visual dashboard scene, layout objects are used to display query results in visual formats, such as charts, graphs, maps, tables and grids. For most layout objects, you display the results of multiple queries in a single layout. For example, you could create a single XY chart that displays sales figures derived from one query and spending figures derived from another query. You can nest multiple layout objects. When you nest layout objects, you can pass query result information from a higher level layout object to a lower level layout object. Because you have the ability to pass this information, you can use nested layout objects to display more detailed information that relates to a specific data value.

Layout Properties dialog

An interface that allows you to set all the options that are available for formatting query results. The Layout Properties dialog uses a tree structure to represent the columns in the query results. Depending on what you have selected in the query results, the Layout Properties tree displays branches for all columns in the query results or for a single column in the query results. You can apply formatting options to entire columns, individual cells, column headings, and summary cells.

Legacy Visionary world

You can create a new dashboard from an existing Visionary world that was created using a previous version of QMF Visionary. The Visionary world that you import must have been exported as an XML file using QMF for Visionary Version 8.1, Fix Pack 13 or later.

LOB data

A large object (LOB) is a Db2 for Linux, UNIX®, and Windows data type that houses nontraditional data such as text, multimedia, image, video, photograph, sound, or any very large data file inside a database table. Retrieving or saving LOB data can consume a substantial amount of resources.

network repositories

A connection to a shared repository using direct network access. Network connectivity permits users to access a shared repository without the need for direct database connectivity from the workstation. Connecting to repositories using a network connection allows for consolidation of data access at a

central server and removes the need to distribute JDBC drivers across each desktop running QMF for Workstation. This type of connection is not applicable to QMF for WebSphere users.

OLAP queries

OLAP queries enable users to dynamically access different dimensions of multidimensional data. Using QMF for Workstation and QMF for WebSphere you can create an OLAP query for multidimensional data sources that is queried using MDX and accessed using XMLA.

palette objects

From the Palette view, you can insert any graphical object that is listed into your visual project. All objects from the Palette view can be inserted in a visual dashboard. Not all objects available from the Palette view can be inserted in a visual report. The objects that cannot be inserted in a visual report have interactive properties, and visual reports are not interactive.

personal repositories

Personal repositories serve a single user and are stored in the personnel settings directory of the workstation (server) that is running the QMF for Workstation/WebSphere application interface.

perspectives

An interface used for querying data and formatting results.

procedures

A set of commands that enable you to run queries, print reports, import and export data, as well as perform other functions.

Project Explorer view

The Project Explorer view details in a tree structural all the elements and objects of each visual report and each dashboard.

Prompted Query editor

An interface that allows you to build simple to complex queries. The Prompted Query editor prompts you for the information it needs to build the SQL statement. You can also set sort and row conditions using the Prompted Query editor. (QMF for Workstation only)

QMF catalogs

A set of database tables that contain saved objects (queries, procedures and forms); user resource limits and profiles; reports; and other miscellaneous settings and information. QMF catalogs reside on database servers that host a Db2 database.

Query Editor

An interface that allows you to open any database table that is accessible to you in your workspace.

query parameters

Query parameters contain the value that will be sent to the query and used at runtime.

regular save

When saving your query results data using a "regular save" method, the interface that you are using (QMF for Workstation or QMF for WebSphere) saves the retrieved query results to the database using an individual SQL INSERT statement for each row of data. This type of save can be expensive in terms of performance due to the overhead of passing large amounts of data back to the database.

relational query

A query is a request for information from a data source. To request information from a relational data source your query is constructed using SQL statements.

resource limits

QMF for Workstation/WebSphere resource limits control your data source access and resource consumption. You must be connected to the data source in order to view the resource limits that are in effect for your user ID when you access that data source.

scene parameters

A parameter that is available only to a specific scene in a visual dashboard. You can pass scene parameters when you create embedded scenes and event actions.

scheduled tasks

You can schedule your procedure, query, quick report, or visual report to run at any given time using your local scheduler or the QMF for WebSphere scheduler. Objects that have been scheduled to run at another time are referred to as scheduled tasks.

shared repositories

Shared repositories are located on a database and can be shared by many users. Only QMF for Workstation/WebSphere administrators can create shared repositories.

SQL Query editor

For those with SQL experience, one way of creating a query is to type their own SQL statements in the SQL Query editor. You can write a single SQL statement that will return a single result set or multiple SQL statements that will return multiple result sets. The editor provides coloring support for your SQL statements.

static SQL packages

Several tables in repository storage are used by QMF to store processing information. Some of these tables store sensitive information, for example the permissions table. By default all users can access these repository storage tables and make changes to the tables. You can choose to protect the repository storage tables. In protection mode, the repository storage tables are accessed using a collection of stored procedures or static SQL packages depending on what the database that is hosting the repository storage supports. Users of the repository storage must then be granted permission to run the stored procedures or static SQL packages.

substitution variables

Substitution variables are used to enter changing values to a SQL query at run time. This feature enables you to substitute a part of an SQL statement and make it more generic. Substitution variables are active only while the object (query, procedure or form) is running. As a result, only one object can access the substitution variable. The variable will not exist after the object is executed.

Table Editor

An interface that allows you to open any database table that is accessible to you in your workspace.

Table Viewer

An interface that allows you to open any database table that is accessible to you in your workspace.

translation tables

Translation tables enable dashboard designers to publish a single visual dashboard that may be viewed in multiple languages. When content developers create a dashboard, button captions, labels, tool tips and other embedded text is entered directly into the property values of the dashboard objects. Using translation tables, content developers can supply alternate versions of this entered text, to be displayed when the user is in a different locale. For example, a Japanese developer may elect to author a new dashboard in Japanese but then also include translation tables for German and French, each of which will be used when the dashboard is viewed under German and French locales.

usage codes

Usage codes provide summary information about the data in a column. For example, usage codes can provide total summary information at the end of a column, or partial summaries at control breaks in a table. The usage codes available depend on the data in the column and the type of summary.

user preferences

The **Preferences** dialog is used to set user preferences that will apply to specific functions in the current QMF for Workstation/WebSphere application interface session. The **Preferences** dialog consists of two panes. The left pane displays a hierarchical tree to organize the preferences and the right pane displays the pages of actual preference values.

variables bar

You can use the Variables bar to specify a property value. The Variables bar is located in the menu bar. From the Variables bar you can choose a global or local parameter that has been defined for the visual project and a column name from a set of query results.

views

QMF for Workstation/WebSphere views support editors and provide alternative presentations as well as ways to navigate the information in your window. Each view displays as a pane in the main workstation window. Each perspective of QMF for Workstation/WebSphere initially displays the views that are most commonly used to perform specific functions.

visual dashboards

Visual dashboards present interactive or persistent data obtained from querying multiple heterogeneous data sources across the enterprise. Visual dashboards present data in a scene format using a wide variety of graphics including charts, graphs, maps and user interface widgets. Data driven graphical objects can be easily linked so that user selections will trigger unique data displays. Content developers can create a visual dashboard that can be viewed by multiple users with either QMF for Workstation or QMF for WebSphere.

visual designer

You can quickly design visual reports and dashboards using the intuitive Visual Designer perspective that includes an editor that presents both design and runtime views of the project; the Project Explorer view

which details the structural content of each visual report and each dashboard; and the Palette, Properties, Events and Output views which support the Visual Designer editor and assist in creating visual reports and dashboards.

visual reports

Visual reports are page-based, printable reports that include both formatted text and graphics to display persistent data to a wide variety of users. Visual reports can also contain data driven graphics (such as maps and charts) inserted in different sections (such as the headers or footers) of the report. Each of the data driven graphics can present data from multiple queries that are run across the enterprise.

web service repositories

You create a Web service repository connection to set up the information that enables you to connect to a shared repository through QMF for WebSphere web services API using a HTTP or HTTPS connection. Web-based connectivity permits users to access a shared repository without the need for direct database connectivity from the workstation. Connecting to repositories using a web-based connection allows for consolidation of data access at a central server and removes the need to distribute JDBC drivers across each desktop running QMF for Workstation. This type of connection does not apply to QMF for WebSphere.

workspaces

All of the data sources and objects that you can access are contained in one or more workspaces that have been pre-populated for you by the QMF for Workstation/WebSphere administrator. Each workspace to which you have access is listed in the Workspaces view. From the Workspaces view, you can perform the majority of query and reporting functions.

zoom

If you are using QMF for Workstation, you can use the Zoom option to view in hexadecimal or binary number the contents of a cell in the query results.

Index

Special Characters

"&variablename=variablevalue" [534](#)
/Batch [534](#)
/IDisplay [534](#)
/IObject:"ObjOwner"."ObjName" [534](#)
/IPassword:password [534](#)
/IServer:servername [534](#)
/IUserID:userID [534](#)
/RConnection:connectionname [534](#)
/RDBUser:rdbuserID [534](#)
/RObject:pathtobject [534](#)
/RPassword:rpassword [534](#)
/Run [534](#)
/RUser:ruserID [534](#)
&&Variable [142](#), [144](#), [158](#)
&an [260](#)
&CALCid [260](#)
&COUNT [260](#)
&DATE [260](#)
&Global Variables [260](#)
&HTML Variables [260](#)
&n [260](#)
&PAGE [260](#)
&ROW [260](#)
&TIME [260](#)

A

about extended layout objects [459](#)
about resource limits [537](#)
accessibility
 assistive technology [546](#)
 keyboard equivalents [545](#)
 keyboard navigation [545](#)
 navigation in WebSphere [548](#)
 navigation in Workstation [546](#)
 operating system [545](#)
ACROSS [261](#)
ACTION [150](#), [164](#)
actions
 action groups [236–238](#)
 execute object
 export query [226](#)
 export query to mail [227](#)
 export quick report [230](#)
 export quick report to mail [229](#)
 export visual report [232](#)
 export visual report to mail [231](#)
 print query [229](#)
 resource manipulation
 copy [234](#)
 send mail [235](#)
ad hoc reports [432](#)
Ad hoc reports
 creating [347](#)
Add Calculated Column [214](#)

Add to Favorites [20](#)
Add to Startup [20](#)
adding an Execute JavaScript event
 how to [370](#)
adding objects [432](#)
Adding objects
 Canvas
 HTML5 mode [433](#), [434](#)
Administrator perspective [1](#), [3](#)
aggregations [325](#)
analysis paths [329](#)
Analytical queries
 append queries [95](#)
 conditional grouping [110](#)
 crosstab [105](#)
 except queries [99](#)
 intersect queries [101](#)
 join [103](#)
 normalize [107](#)
 union queries [97](#)
Analytics [145](#)
Analytics objects
 opening [529](#)
 opening with a source [529](#)
 opening without a source [530](#)
 procedure commands
 DISPLAY [530](#)
 EXPORT [530](#)
 PRINT [530](#)
ANCHORn [252](#)
Appearance preferences [9](#)
append queries [95](#)
applying
 conditional format [273](#), [281](#)
 conditional formatting [273](#), [281](#)
ATTACHMENT [152](#)
AutoFit [214](#)
Average [214](#)
AVERAGE [261](#)
axes chart format options [317](#)

B

best fit models [287](#)
binary [213](#)
binding resource limits [541](#)
BODY [152](#)
BOLD [155](#)
Boolean [355](#)
Border [361](#)
BOTTOM [140](#)
BOTTOMMARGIN [155](#)
Break Footing [260](#)
Break Heading [260](#)
Breakn [261](#)
BREAKnX [261](#)
breakpoints [480](#)

- Breaks formatting options
 - Blank lines after footing [250](#)
 - Blank lines after heading [250](#)
 - Blank lines before footing [250](#)
 - Blank lines before heading [250](#)
 - Break footing text [250](#)
 - Break heading text [250](#)
 - New page for break [250](#)
 - New page for footing [250](#)
 - Put break summary at line [250](#)
 - Repeat detail heading [250](#)
- building prompt hierarchy filter [190](#)

C

- cache resource limits [544](#)
- caching
 - repository data [42](#)
- CALCid [261](#)
- calculated columns [190–192](#), [394](#)
- calculated expressions [406](#)
- Calculation Expression [260](#)
- Calculations formatting options
 - Edit [253](#)
 - Expression [253](#)
 - ID [253](#)
 - Pass nulls? [253](#)
 - Width [253](#)
- canvas
 - editing linked objects [437](#)
 - linked objects [437](#)
- Canvas
 - types [432](#)
- Canvas mode [401](#)
- Category Axis [325](#)
- category chart format options [316](#)
- CCLIST [152](#)
- CCSID [145](#), [213](#)
- Change favorite name [27](#)
- Change Startup Name [27](#)
- character data edit codes [517](#)
- CHARSET [155](#)
- chart area chart format options [317](#)
- charts
 - creating [313](#)
 - creating from query results grid [322](#)
 - creating in Display editor [323](#)
 - editing [326](#)
 - specifying chart format [315](#)
 - specifying chart types [314](#)
 - specifying data series [314](#)
- Cheat Sheets view [3](#)
- classic reports [245](#)
- client area chart format options [322](#)
- Close dialog [384](#)
- Color [355](#)
- color maps
 - creating [480](#)
 - using [481](#)
- color sequences
 - creating [478](#)
 - using [479](#)
- Column Expression [260](#)
- Columns formatting options

- Columns formatting options (*continued*)
 - Edit [253](#)
 - Heading [253](#)
 - Indent [253](#)
 - Num [253](#)
 - Seq [253](#)
 - Usage [253](#)
 - Width [253](#)
- Columns with aggregates [324](#)
- combo box [420](#), [422](#)
- command bar [535](#)
- command line
 - parameters [534](#)
- CommandLine [145](#)
- COMMENT [150](#), [164](#)
- comments, sending to IBM [xv](#)
- compile
 - visual dashboard [395](#)
 - visual report [395](#)
- compound condition filter [188](#)
- Condition [260](#)
- conditional grouping [110](#)
- Conditions formatting options
 - Expression [255](#)
 - ID [255](#)
 - Pass nulls [255](#)
- CONFIRM [144](#), [145](#), [150](#), [158](#), [164](#)
- CONNECT
 - PASSWORD [141](#)
 - ServerName [141](#)
 - UserName [141](#)
- connecting to data [29](#)
- Connectivity [408](#)
- connectivity tool [412](#)
- connector objects
 - Connection Point [427](#)
 - Elbow Connector [427](#)
 - Spline Connector [427](#)
 - Straight Connector [427](#)
- Connectors [408](#)
- container objects
 - Canvas [429](#)
 - Embedded Content [429](#)
 - Embedded Scene [429](#)
- Containers [408](#)
- content assist [59](#)
- control objects
 - Button [419](#)
 - Check Box [419](#)
 - Combo Box [419](#)
 - Date Time [419](#)
 - Horizontal Slider [419](#)
 - List Box [419](#)
 - Prompt Hierarchy [419](#)
 - Radio Group [419](#)
 - Slicer [419](#)
 - Text Box [419](#)
 - Tree [419](#)
 - Vertical Slider [419](#)
- Controls [408](#)
- conventions for highlighting [xiii](#)
- convert
 - CSV [210](#)
 - DBF [210](#)

- convert (*continued*)
 - IXF [210](#)
 - TXT [210](#)
 - XML [210](#)
- CONVERT
 - QueryName [141](#)
- Convert to HTML [284](#)
- Convert to Visual Report [284](#)
- COPIES [155](#)
- Copy [20](#)
- copying
 - tables [125](#)
- copying tables [125](#)
- correct format [46](#)
- Count [214](#)
- COUNT [261](#)
- CPCT [261](#)
- Create
 - JavaScript Tables [495](#)
- Create Cyclicity [287](#)
- Create File [380](#)
- Create New Visual Dashboard wizard
 - tabs
 - Local [351](#)
 - Predefined [351](#)
 - Repository [351](#)
- Create New Visual Report wizard
 - Local tab [338](#)
 - Predefined tab [338](#)
 - Repository tab [338](#)
- Create Performance [287](#)
- Create Seasonality [287](#)
- Create Trend [287](#)
- creating [329](#)
- crosstab queries [105](#)
- CSUM [261](#)
- cube data [87](#)
- Cube Structure tree [87](#), [88](#)
- cumulative forecast error [301](#)
- Cumulative Percentage of Group [214](#)
- Cumulative Percentage of Total [214](#)
- Cumulative Sum [214](#)
- curve fitting [299](#)
- Curve fitting methods [287](#)
- Custom [408](#)
- custom functions [192](#)
- customizing perspectives [7](#)

D

- data
 - accessing [29](#)
 - connecting to [29](#)
- data source connections [353](#)
- Data Source Connections view [3](#)
- Data Source Object [284](#)
- data source options
 - embedded [289](#)
 - linked [289](#)
- data sources
 - listing objects [247](#)
 - selecting new [127](#)
- data symbol objects
 - Candlestick [426](#)

- data symbol objects (*continued*)
 - Event Band [426](#)
 - Horizontal Label Bar [426](#)
 - Horizontal Value Bar [426](#)
 - Marker [426](#)
 - Price Interval Bar [426](#)
 - Threshold Band [426](#)
 - Vertical Label Bar [426](#)
 - Vertical Value Bar [426](#)
- Data Symbols [408](#)
- data templates
 - visibility [469](#)
- data types [406](#)
- Database folder [19](#)
- DATAFORMAT [145](#)
- date data edit codes [518](#)
- date parameters
 - date column [291](#)
 - from [291](#)
 - number of future periods [291](#)
 - period [291](#)
- DateTime [355](#)
- DATETIME [155](#)
- DateTimeSet [355](#)
- Debugging visual dashboards [493](#)
- Default editor [117](#)
- Delete [20](#)
- Design mode [401](#)
- detail [180](#)
- Detail Block [260](#)
- Detail Heading [260](#)
- Details formatting options
 - Blank lines after block [255](#)
 - Detail block text [255](#)
 - Detail heading text [255](#)
 - Enable [255](#)
 - Include column headings with detail headings [255](#)
 - Keep block on page [255](#)
 - New page for detail block [255](#)
 - Put tabular data at line [255](#)
 - Repeat detail heading [255](#)
- dialog templates
 - background [385](#)
 - border [385](#)
 - content paddings [386](#)
 - Nine Patch [386](#)
 - scene background [387](#)
 - shadow [387](#)
 - title background [388](#)
 - title font [388](#)
 - title frame [388](#)
- directly importing an IXF file into a database table [207](#)
- DISPLAY
 - &&Variable [142](#)
 - ObjectName [142](#)
- Display editor [323](#)
- Display editor panes
 - Display modes [323](#)
 - Layout structure [323](#)
 - Result set [323](#)
- Display Excel Sheet [214](#)
- display modes
 - editing chart format
 - axes [317](#)

- display modes (*continued*)
 - editing chart format (*continued*)
 - category [316](#)
 - chart area [317](#)
 - client area [322](#)
 - gridlines [319](#)
 - labels [319](#)
 - legend [321](#)
 - plot area [321](#)
 - range [320](#)
 - series [315](#)
 - title [320](#)
 - value(Y)series [316](#)
 - X-Axis [318](#)
 - Y-Axis [318](#)

- Display modes pane [323](#), [324](#)

- Display Report [214](#)

- Display Report wizard [245](#)

- display settings

- Columns [410](#)

- Details [410](#)

- Icons Only [410](#)

- List [410](#)

- double exponential smoothing [298](#)

- Double Exponential Smoothing [287](#)

- DRAW

- TableName [143](#)

- TYPE [143](#)

- Draw Query editor

- listing data source tables [75](#)

- Draw Query wizard [56](#)

- Drill Down [214](#)

- Drill Up [214](#)

- Drill-down Path display modes [331](#)

- Drill-down Paths

- analysis paths [331](#)

- display modes [331](#)

- drilldown [358](#)

- Drilldown wizard [358](#)

- driver class name [13](#), [33](#)

- DSQAO [511](#)

- DSQCP [515](#)

- DSQDC [515](#)

- DSQEC [513](#)

- DSQW [506](#)

- Dynamart Filter Wizard [187](#)

- dynamic SQL [1](#)

E

- Eclipse [1](#)

- Edit

- JavaScript Tables [499](#)

- EDIT

- &&Variable [144](#)

- ObjectName [144](#)

- edit codes

- character data [517](#)

- custom codes for classic reports [523](#)

- date data [518](#)

- graphic data [518](#)

- metadata [527](#)

- numeric data [519](#)

- samples [523](#)

- edit codes (*continued*)

- time data [521](#)

- timestamp data [521](#)

- user defined [522](#)

- editing database tables [124](#)

- editor window [401](#)

- editors

- Prompted Query [56](#)

- Query Builder [56](#)

- SQL [56](#)

- email

- query results [197](#)

- embedded scenes

- parameterized [362](#)

- End Session [379](#)

- ERASE

- CONFIRM [144](#)

- FOLDER [144](#)

- ObjectName [144](#)

- event actions

- assigning [363](#)

- Close dialog [390](#)

- Create File [380](#)

- End Session [379](#)

- Execute JavaScript [370](#)

- Execute procedure [363](#), [369](#)

- Execute shell command [363](#), [372](#)

- Execute SQL statement [363](#), [368](#)

- Export to excel [376](#)

- Export to Excel [363](#)

- Export to PDF [379](#)

- Invalidate Query Cache [382](#)

- Jump to new location [363](#), [364](#)

- Launch LOB [383](#)

- Load Cookie [392](#)

- Modify Data [373](#)

- Navigate back [363](#), [367](#)

- Navigate to an embedded scene [366](#)

- Navigate to embedded scene [363](#)

- Open URL [391](#)

- Print scene [363](#), [370](#)

- Refresh Component [382](#)

- Refresh geolocation [392](#)

- Reload scene [378](#)

- Save Cookie [391](#)

- Send Email [371](#)

- Send Event to Embedded Scene [371](#)

- Set focus [378](#)

- Set values [363](#), [366](#)

- Show dialog [389](#)

- Show message [363](#), [375](#)

- EventInterval [413](#)

- events

- about [366](#)

- ExternalEvent events [366](#)

- Events view [3](#), [407](#)

- except queries [99](#)

- EXECUTE

- CommandLine [145](#)

- Execute JavaScript [363](#), [370](#)

- Execute procedure [363](#), [369](#)

- Execute shell command [363](#), [413](#)

- Execute SQL statement [363](#), [368](#), [413](#)

- expiration schedules

expiration schedules (*continued*)

- daily [487](#)
- hourly [487](#)
- monthly [488](#)
- specific dates [488](#)
- weekly [487](#)

Explore [20](#)

export

- Analytics objects [531](#)
- chart [327](#)
- forecast [309](#)
- query results
 - database [198](#)
 - file [200](#)
 - file export options [201](#)
- reports [247, 346](#)

Export [214](#)

EXPORT

- CCSID [145](#)
- CONFIRM [145](#)
- DATAFORMAT [145](#)
- FileName [145](#)
- LANGUAGE [145](#)
- LENGTH [145](#)
- LOBFILE [145](#)
- LOBSINFILE [145](#)
- LOBSTO [145](#)
- MODE [145](#)
- ObjectName [145](#)
- ORDER [145](#)
- OUTPUTMODE [145](#)
- SEPARATOR [145](#)
- SPLIT [145](#)
- UNICODE [145](#)
- UNITS [145](#)
- WIDTH [145](#)

Export images [400](#)

export query operation [226](#)

export query to mail operation [227](#)

export quick report operation [230](#)

export quick report to email operation [229](#)

Export to excel [376](#)

Export to Excel [363](#)

Export to PDF [379](#)

export visual report operation [232](#)

export visual report to mail operation [231](#)

expression designer [187, 404](#)

expressions

- column names [405](#)
- constants [405](#)
- elements [405](#)
- examples [407](#)
- formatting symbols [405](#)
- functions [405](#)
- operators [405](#)
- syntax rules [406](#)

ExternalEvent events [366](#)

ExternalEventID [371](#)

F

fast save [218](#)

Favorites folder [26](#)

feedback, sending to IBM [xv](#)

file formats

- HTML [247, 346](#)
- PDF [247, 346](#)
- text [247, 346](#)

file objects [53](#)

file-based repositories [35, 38](#)

FileName [145, 150](#)

FilePath [355](#)

filter page [87](#)

Filter tab [90](#)

filtering objects

- Canvas [436](#)

filtering query results [187](#)

Final formatting options

- Blank lines before text [258](#)
- New page for final text [258](#)
- Put final summary at line [258](#)
- Text [258](#)

Final Text [260](#)

Find [214](#)

First [214](#)

FIRST [261](#)

Font [214](#)

FontName [355](#)

forecast chart properties [306](#)

forecast cyclicity [305](#)

forecast grid properties [307](#)

forecast performance

- cumulative forecast error [301](#)
- mean absolute deviation [301](#)
- mean absolute percent error [301](#)
- mean square error [301](#)
- tracking signal [301](#)

forecast seasonality

- global average [304](#)
- moving average [304](#)

forecast trends

- global trend [302](#)
- linear trend [302](#)
- local trend [302](#)

forecasting models options

- Curve fitting [295, 299](#)
- Double exponential smoothing [295, 298](#)
- Holt-Winters method [295, 298](#)
- Multiplicative decomposition [295, 299](#)
- Neural network [295, 300](#)
- Polynomial regression [295, 300](#)
- Simple moving average [295, 297](#)
- Single exponential smoothing [295, 297](#)
- Weighted moving average [295, 297](#)

forecasts

- chart properties [306](#)
- cyclicity [305](#)
- data source options [289](#)
- date parameters [291](#)
- forecasting models options [295, 297–300](#)
- grid properties [307](#)
- grouping hierarchy options [292](#)
- overview [287](#)
- performance [301](#)
- seasonality [304](#)
- strategy options [294](#)
- trends [302](#)

FORM [155, 158](#)

Form menu commands

- Convert to HTML [284](#)
- Convert to Visual Report [284](#)
- Data Source Object [284](#)
- Set Data Source [284](#)
- Set Font [284](#)
- Set User Information [284](#)

Form structure tree

- Breaks [248](#)
- Calculations [248](#)
- Columns [248](#)
- Conditions [248](#)
- Details [248](#)
- Final [248](#)
- Main [248](#)
- Options [248](#)
- Page [248](#)

form variables

- &n [260](#)
- &CALCid [260](#)
- &COUNT [260](#)
- &DATE [260](#)
- &Global Variables [260](#)
- &HTML Variables [260](#)
- &n [260](#)
- &PAGE [260](#)
- &ROW [260](#)
- &TIME [260](#)

Format [214](#)

formatting options

breaks

- Blank lines after footing [250](#)
- Blank lines after heading [250](#)
- Blank lines before footing [250](#)
- Blank lines before heading [250](#)
- Break footing text [250](#)
- Break heading text [250](#)
- New page for break [250](#)
- New page for footing [250](#)
- Put break summary at line [250](#)
- Repeat detail heading [250](#)

calculations

- Edit [253](#)
- Expression [253](#)
- ID [253](#)
- Pass nulls? [253](#)
- Width [253](#)

columns

- Edit [253](#)
- Heading [253](#)
- Indent [253](#)
- Num [253](#)
- Seq [253](#)
- Usage [253](#)
- Width [253](#)

conditions

- Expression [255](#)
- ID [255](#)
- Pass nulls [255](#)

details

- Blank lines after block [255](#)
- Detail block text [255](#)
- Detail heading text [255](#)
- Enable [255](#)

formatting options (*continued*)

details (*continued*)

- Include column headings with detail headings [255](#)
- Keep block on page [255](#)
- New page for detail block [255](#)
- Put tabular data at line [255](#)
- Repeat detail heading [255](#)

final

- Blank lines before text [258](#)
- New page for final text [258](#)
- Put final summary at line [258](#)
- Text [258](#)

main

- Edit [249](#)
- Heading [249](#)
- Indent [249](#)
- Seq [249](#)
- Total width of report columns [249](#)
- Usage [249](#)
- Width [249](#)

options

- Across heading separator [257](#)
- Across summary column [257](#)
- Automatic reordering of report columns [257](#)
- Break summary separator [257](#)
- Column heading separator [257](#)
- Column wrapped lines kept on a page [257](#)
- Default break text (*) [257](#)
- Detail spacing [257](#)
- Final summary separator [257](#)
- Function name in column heading when grouping [257](#)
- Line wrapping width [257](#)
- Number of fixed columns in report [257](#)
- Outlining for break columns [257](#)
- Page renumbering at the highest break level [257](#)
- Place LOB contents inline [257](#)
- Report text line width [257](#)

page

- Blank lines after footing [259](#)
- Blank lines after heading [259](#)
- Blank lines before footing [259](#)
- Blank lines before heading [259](#)
- Page footing text [259](#)
- Page heading text [259](#)

forms

- designing [248](#)
- saving with data source [264](#)
- saving without data source [264](#)

FORWARD [150](#)

free style condition filter [189](#)

Frequently Used [408](#)

From [152](#)

From data source [246](#)

From file [246](#)

From open object [246](#)

From repository [246](#)

full outer join [63](#), [69](#)

functions.js [192](#)

G

General preferences [8](#)

getting started [1](#)

- global average [304](#)
- global parameters
 - adding [485](#)
 - ExternalEventID [484](#)
 - SendExternalEvent [484](#)
 - Translation Table [484](#)
 - using [485](#)
 - ViewerHeight [484](#)
 - ViewerWidth [484](#)
 - ViewerX [484](#)
 - ViewerY [484](#)
 - ViewerZoom [484](#)
- Global parameters [404](#)
- global resources
 - color maps
 - creating [480](#)
 - using [481](#)
 - color sequences
 - creating [478](#)
 - using [479](#)
 - expiration schedules
 - daily [487](#)
 - hourly [487](#)
 - monthly [488](#)
 - specific dates [488](#)
 - weekly [487](#)
 - global parameters
 - adding [485](#)
 - using [485](#)
 - images [477](#)
 - parameters [477](#)
 - security lists [477](#), [481](#)
 - stock images
 - creating [482](#)
 - using [483](#)
 - translation tables
 - switching [490](#)
- global trend [302](#)
- global variables
 - DSQAO [511](#)
 - DSQCP [515](#)
 - DSQDC [515](#)
 - DSQEC [513](#)
 - DSQQW [506](#)
 - system [11](#), [506](#)
 - user [10](#), [505](#)
- Go to [214](#)
- Google Maps
 - Google Map Marker object [472](#), [473](#)
 - Google Map object [471](#)
 - Google Map Object [476](#)
 - Google Map Polygon object [475](#)
 - Google Map Polyline object [474](#)
- Grab [408](#)
- grand total [180](#)
- graphic data edit codes [518](#)
- gridlines chart format options [319](#)
- GROUP [261](#)
- grouping and aggregation
 - fields [195](#)
- Grouping and Aggregation
 - Average [214](#)
 - Count [214](#)
 - Cumulative Percentage of Group [214](#)

- Grouping and Aggregation (*continued*)
 - Cumulative Percentage of Total [214](#)
 - Cumulative Sum [214](#)
 - First [214](#)
 - Last [214](#)
 - Maximum [214](#)
 - Minimum [214](#)
 - No Aggregation [214](#)
 - Percentage of Group [214](#)
 - Percentage of Total [214](#)
 - Side Group [214](#)
 - Standard Deviation [214](#)
 - Sum [214](#)
 - Top Group [214](#)
- grouping hierarchy options
 - available columns [292](#)
 - selected columns [292](#)

H

- header [180](#)
- help content [12](#)
- Help preferences [11](#)
- Help view [3](#)
- hexadecimal [213](#)
- Hierarchical filter [90](#)
- highlighting conventions [xiii](#)
- Holt-Winters method [287](#), [298](#)
- host [12](#)
- HP-UX [1](#)
- HTML [401](#)
- HTML tables [242](#)
- HTML variable [252](#)

I

- ICatalogPassword:catalogpassword [534](#)
- ICatalogUserID:cataloguserID [534](#)
- icons
 - Create Cyclicity [287](#)
 - Create Performance [287](#)
 - Create Seasonality [287](#)
 - Create Trend [287](#)
 - Run Query [287](#)
 - Show Forecast Outline Tree [287](#)
- IDENTIFIER [143](#)
- IMAGEn [252](#)
- images [477](#)
- import
 - CSV
 - result set structure [205](#)
 - separator options [204](#)
 - legacy Visionary world [353](#)
 - legacy visual reports [340](#)
 - table data [209](#)
 - tables [209](#)
 - TXT
 - result set structure [205](#)
 - separator options [204](#)
 - XLS [207](#)
 - XLSX [207](#)
- Import [45](#)
- IMPORT

- IMPORT *(continued)*
 - [ACTION 150](#)
 - [COMMENT 150](#)
 - [CONFIRM 150](#)
 - [FileName 150](#)
 - [LANGUAGE 150](#)
 - [LOBSFROM 150](#)
 - [ObjectName 150](#)
 - [SHARE 150](#)
- import CSV
 - [result set structure 205](#)
 - [separator options 204](#)
- import settings [45](#)
- Import settings from QMF for Windows [45](#)
- import TXT
 - [result set structure 205](#)
 - [separator options 204](#)
- import XLS [207](#)
- import XLSX [207](#)
- importing data contained in ixf files [206](#)
- Informix [1](#)
- inner join [63](#), [69](#)
- inserting [420](#)
- Integer [355](#)
- intersect queries [101](#)
- Invalidate Query Cache [382](#)
- ITALIC [155](#)
- iterator [238](#)

J

- JARS [13](#), [33](#)
- JavaScript
 - [functions 397](#)
 - [module 397](#)
- JavaScript functions
 - [proc.exec\(\) 171](#)
 - [proc.exists\(\) 172](#)
 - [proc.getCellValue\(\) 173](#)
 - [proc.getColumnName\(\) 173](#)
 - [proc.getColumnType\(\) 174](#)
 - [proc.getNumColumns\(\) 174](#)
 - [proc.getNumRows\(\) 174](#)
 - [proc.getVariable\(\) 172](#)
 - [proc.include\(\) 175](#)
 - [proc.messageBox\(\) 175](#)
 - [proc.prompt\(\) 176](#)
 - [proc.setCellValue\(\) 177](#)
- JavaScript modules [192](#), [477](#)
- JavaScript Modules [398](#)
- JavaScript procedures
 - [functions 171](#)
- JavaScript Tables [495–497](#)
- JDBC [1](#)
- JDBC drivers [1](#), [32](#)
- JDBC libraries [13](#), [33](#)
- JDBC library settings [32](#)
- JDBC preferences [13](#), [33](#)
- join queries [103](#)
- Jump to new location [363](#), [364](#), [413](#)

L

- labels chart format options [319](#)
- LANGUAGE [145](#), [150](#), [164](#)
- Last [214](#)
- LAST [261](#)
- Launch LOB [383](#)
- Layers view [3](#), [477](#)
- layout objects
 - Bar chart [438](#)
 - Bar Chart
 - [extended 443](#)
 - [OLAP 444](#)
 - Candlestick chart [438](#)
 - Candlestick Chart [449](#)
 - Cluster graph [438](#)
 - Cluster Graph [450](#)
 - Column chart [438](#)
 - Column Chart
 - [extended 443](#)
 - [OLAP 444](#)
 - data symbols
 - [graphic objects 463](#)
 - data templates
 - [new 466](#)
 - default connectors [462](#)
 - Dial [457](#)
 - editing a solid chart [458](#)
 - Event Band chart [438](#)
 - Event Band Chart [450](#)
 - [extended 459](#)
 - [extending functionality 459](#)
 - Grid [438](#), [442](#)
 - Horizon [438](#)
 - Linear map [438](#)
 - Linear Maps [451](#)
 - Matrix [438](#), [452](#)
 - [multiple query results 464](#)
 - Multivariate chart [438](#)
 - Multivariate Chart [447](#)
 - [nesting 470](#)
 - [OLAP data 460](#)
 - Organization chart [438](#)
 - Organization Chart [453](#)
 - Pie chart [438](#)
 - Pie Chart
 - [extended 445](#)
 - [OLAP 446](#)
 - [queries 458](#)
 - [query parameters 458](#)
 - Scatter chart [438](#)
 - Scatter Chart [447](#)
 - Simple form [438](#)
 - Simple Form [451](#)
 - Solid chart [438](#), [457](#)
 - Spiral [438](#), [452](#)
 - Stock chart [438](#)
 - Stock Chart [448](#)
 - Table
 - [populated 439](#)
 - [simple 439](#)
 - [summary row formatting 441](#)
 - [summary rows 440](#)
 - Timeline chart [438](#)

layout objects (*continued*)

- Timeline Chart [449](#)
- Tree chart [438](#)
- Tree Chart [453](#)
- XY chart [438](#)
- XY Chart [447](#)
- Layout page [87](#)
- layout properties
 - Conditional [180](#)
 - Font [180](#)
 - Format [180](#)
 - General [180](#)
- Layout Properties dialog
 - conditional formatting options [186](#)
 - font fields [183](#)
 - font formatting options [182](#)
 - format fields [184](#)
 - formatting options [184](#)
 - general column formatting [182](#)
- Layout structure pane [323](#), [325](#)
- Layout Structure tree [195](#)
- Layout tab [88](#)
- Layout view [89](#)
- layout wizard [410](#)
- Layouts [408](#)
- left outer join [63](#), [69](#)
- LEFTMARGIN [155](#)
- legacy Visionary worlds
 - importing [353](#)
- legacy visual reports
 - importing [340](#)
 - opening [340](#)
- legend chart format options [321](#)
- LENGTH [145](#), [155](#)
- LIMIT LOCAL
 - Value [151](#)
 - VariableName [151](#)
- limits resource limits [538](#)
- linear trend [302](#)
- linking [422](#)
- LINKn [252](#)
- links
 - non-IBM Web sites [622](#)
- Linux [1](#)
- list box [420](#), [422](#)
- Literal [355](#)
- literals [406](#)
- Load Cookie [392](#)
- LOB data [211](#)
- LOB options resource limit
 - overriding [212](#)
- LOB options resource limits [543](#)
- LOBFILE [145](#)
- LOBs preferences [14](#)
- LOBSFROM [150](#)
- LOBSINFILE [145](#)
- LOBSTO [145](#)
- Local parameters [404](#)
- local trend [302](#)
- Log preferences [15](#)
- LookAt [xiv](#)

M

- MAIL TO
 - ATTACHMENT [152](#)
 - BODY [152](#)
 - CCLIST [152](#)
 - From [152](#)
 - METHOD [152](#)
 - ObjectName [152](#)
 - SMTP SERVER [152](#)
 - SUBJECT [152](#)
 - TYPE [152](#)
- MAILTOn [252](#)
- Main formatting options
 - Edit [249](#)
 - Heading [249](#)
 - Indent [249](#)
 - Seq [249](#)
 - Total width of report columns [249](#)
 - Usage [249](#)
 - Width [249](#)
- manage data files in dynamarts [211](#)
- manage in dynamart
 - CSV [211](#)
 - DBF [211](#)
 - IXF [211](#)
 - TXT [211](#)
 - XML [211](#)
- map [394](#)
- Mapping table [211](#)
- matching patterns [46](#)
- Maximum [214](#)
- MAXIMUM [261](#)
- MDX page [87](#)
- mean absolute deviation [301](#)
- mean absolute percent error [301](#)
- mean square error [301](#)
- measures [88](#), [89](#)
- menu commands
 - Personal view [27](#)
 - Workspace [20](#)
- messages
 - methods for accessing [xiv](#)
- metadata edit codes [527](#)
- METHOD [152](#), [164](#)
- Microsoft Windows [1](#)
- Minimum [214](#)
- MINIMUM [261](#)
- Miscellaneous [408](#)
- MODE [145](#), [164](#)
- moving average [304](#)
- Moving Average [287](#)
- multiple result sets [214](#)
- multiplicative decomposition [299](#)
- Multiplicative Decomposition method [287](#)

N

- name filter [23](#)
- Navigate back [363](#), [367](#), [413](#)
- Navigate to an embedded scene [366](#)
- Navigate to embedded scene [363](#)
- network connections [38](#)
- network repositories [35](#)

- neural network [300](#)
- Neural Networks [287](#)
- New [20](#)
- No Aggregation [214](#)
- normalize queries [107](#)
- notices
 - legal [621](#)
- Number [355](#)
- NumberSet [355](#)
- numeric data edit codes [519](#)

O

- object list resource limits [544](#)
- object name [23](#)
- object owner [23](#)
- object tracking resource limits [542](#)
- ObjectName [142](#), [144](#), [145](#), [150](#), [152](#), [155](#), [158](#), [164](#), [534](#)
- objects
 - creating [46](#)
 - data source [247](#)
 - file [53](#)
 - inserting [410](#)
 - opening [51](#)
 - palette [409](#)
 - QMF catalog [52](#)
 - repository [53](#)
 - saving
 - in a repository [50](#)
 - in the QMF catalog [51](#)
 - to a file [50](#)
- OLAP folder [19](#)
- OLAP queries
 - building offline [88](#)
 - building online [89](#)
 - filtering cube data [90](#)
 - modifying [88](#)
 - opening [88](#)
 - retrieving cube data [87](#)
- OMIT [261](#)
- Open [20](#)
- Open From [20](#)
- open perspective [7](#)
- Open URL [391](#)
- opening the contents of an IXF file [206](#)
- Options formatting options
 - Across heading separator [257](#)
 - Across summary column [257](#)
 - Automatic reordering of report columns [257](#)
 - Break summary separator [257](#)
 - Column heading separator [257](#)
 - Column wrapped lines kept on a page [257](#)
 - Default break text (*) [257](#)
 - Detail spacing [257](#)
 - Final summary separator [257](#)
 - Function name in column heading when grouping [257](#)
 - Line wrapping width [257](#)
 - Number of fixed columns in report [257](#)
 - Outlining for break columns [257](#)
 - Page renumbering at the highest break level [257](#)
 - Place LOB contents inline [257](#)
 - Report text line width [257](#)
- options resource limits [539](#)
- Oracle [1](#)

- ORDER [145](#)
- ORIENTATION [155](#)
- Outline view [3](#)
- Output view [3](#)
- OUTPUTMODE [145](#)
- overview [1](#)

P

- Page Footing [260](#)
- Page formatting options
 - Blank lines after footing [259](#)
 - Blank lines after heading [259](#)
 - Blank lines before footing [259](#)
 - Blank lines before heading [259](#)
 - Page footing text [259](#)
 - Page heading text [259](#)
- Page Heading [260](#)
- Page Setup
 - printing a chart [328](#)
 - printing a drill-down path [332](#)
 - printing a forecast [310](#)
- PAGENO [155](#)
- palette objects
 - default property value [411](#)
 - inserting [409](#)
- Palette view
 - connectivity tool [412](#)
- connector objects
 - inserting [428](#)
- container objects
 - Canvas [431](#)
 - Embedded Content [430](#)
 - Embedded Scene [430](#)
- control objects
 - Date Time [423](#)
 - inserting [420](#)
 - Prompt Hierarchy [425](#)
 - Slicer [422](#)
 - Tree [424](#)
- custom objects [428](#)
- data symbol objects
 - inserting [427](#)
- display settings [410](#)
- layout objects [438](#)
- primitive objects
 - Alignment Panel [418](#)
 - lines [415](#)
 - Picture [417](#)
 - shapes [416](#)
 - text [415](#)
- Properties view [399](#)
- Timer [413](#)
- parameter hints [60](#)
- parameter types
 - Boolean [355](#)
 - Color [355](#)
 - DateTime [355](#)
 - DateTimeSet [355](#)
 - FilePath [355](#)
 - FontName [355](#)
 - Integer [355](#)
 - Literal [355](#)
 - Number [355](#)

parameter types *(continued)*

NumberSet [355](#)
Percentage [355](#)
Point [355](#)
PointSet [355](#)
PolySet [355](#)
Text [355](#)
TextSet [355](#)

parameters

global [483](#)
query [483](#)
scene [483](#)

PASSWORD [141](#)

Paste [20](#)

Paste Link [20](#)

path [12](#)

PCT [261](#)

PDF [401](#)

Percentage [355](#)

Percentage of Group [214](#)

Percentage of Total [214](#)

Personal view [3](#), [26](#)

Personal view menu commands

Change favorite name [27](#)
Change Startup Name [27](#)
Remove from Favorites [27](#)
Remove from Startup [27](#)

perspectives

Administrator [1](#), [3](#)
closing [6](#)
configuring [7](#)
QMF [3](#)
resetting [8](#)
User [3](#), [6](#)
Visual Designer [3](#), [338](#), [399](#)

plot area chart format options [321](#)

Point [355](#)

PointSet [355](#)

polynomial regression [300](#)

PolySet [355](#)

port [12](#)

preferences

Appearance [8](#), [9](#)
Dashboard Runtime [8](#)
General [8](#)
Global Variables [8](#)
Help [8](#), [11](#)
JDBC [13](#), [33](#)
JDBC Libraries [8](#)
LOBs [8](#), [14](#)
Log [8](#), [15](#)
Server-Side File System
[17](#)
SMTP [16](#)
Third Party Libraries [18](#)
Third-Party Libraries [8](#)
user [8](#)
Visual Designer [8](#), [18](#)

Preview mode [401](#)

primitive objects

Alignment panel [414](#)
Arrow [414](#)
Double Arrow [414](#)
Ellipse [414](#)

primitive objects *(continued)*

Label [414](#)
Line [414](#)
Multiple Polygon [414](#)
Multiple Polyline [414](#)
Picture [414](#)
Polygon [414](#)
Polyline [414](#)
Rectangle [414](#)
Rounded Rectangle [414](#)

Primitives [408](#)

PRINT

BOLD [155](#)
BOTTOMMARGIN [155](#)
CHARSET [155](#)
COPIES [155](#)
DATETIME [155](#)
FORM [155](#)
ITALIC [155](#)
LEFTMARGIN [155](#)
LENGTH [155](#)
ObjectName [155](#)
ORIENTATION [155](#)
PAGENO [155](#)
PRINTER [155](#)
RIGHTMARGIN [155](#)
SIZE [155](#)
TOPMARGIN [155](#)
TYPEFACE [155](#)
USEFORMPS [155](#)
WIDTH [155](#)

print options [216](#), [217](#), [310](#), [328](#), [332](#), [536](#)

print query operation [229](#)

Print scene [363](#), [370](#), [413](#)

PRINTER [155](#)

procedure commands

BOTTOM [140](#)
CONNECT [141](#)
CONVERT [141](#)
DISPLAY [142](#)
DRAW [143](#)
EDIT [144](#)
ERASE [144](#)
EXECUTE [145](#)
EXPORT [145](#)
FORWARD [150](#)
IMPORT [150](#)
interface differences [137](#)
LIMIT LOCAL [151](#)
MAIL TO [152](#)
PRINT [155](#)
repository vs. QMF catalog [139](#)
RESET GLOBAL [158](#)
RUN [158](#)
RUNTSO [161](#)
SAVE [164](#)
SET GLOBAL [166](#)
SET INVISIBLE [167](#)
SET LOCAL [167](#)
SET LOCAL WITH VALUES [168](#)
SET OPTIONS [168](#)
SHOW [169](#)
syntax [137](#)
syntax diagrams [138](#)

- procedure commands (*continued*)
 - USE REPOSITORY [170](#)
- procedures
 - add JavaScript conditions [136](#)
 - creating [133](#)
 - edit flowcharts [136](#)
 - flowcharts [135](#)
 - JavaScript [134](#)
- Profiling visual dashboards
 - Analyzer view [491](#)
 - Profiler view [491](#)
- Progress view [3](#)
- Project Explorer [399](#), [400](#)
- Project Explorer folders
 - Connections [337](#), [348](#)
 - Globals [337](#), [348](#)
 - Main Page [337](#)
 - Queries [337](#), [348](#)
 - Scenes [348](#)
- Project Explorer view [3](#), [337](#)
- prompt
 - shared prompts [84](#)
 - simple prompts [76](#)
 - SQL prompts [77](#)
- prompt hierarchies
 - adding to queries [81](#)
 - creating [80](#)
 - embedding to queries [82](#)
 - working with [80](#)
- prompt values
 - saving [79](#), [84](#), [85](#)
- Prompted Query editor
 - adding tables [62](#)
 - columns [60](#)
 - join conditions [60](#)
 - joining columns [64](#)
 - joining tables [63](#)
 - query results [64](#), [71](#)
 - row conditions [60](#), [66](#)
 - sort conditions [60](#), [65](#)
 - tables [60](#)
- properties
 - data types [406](#)
 - expression designer [404](#)
 - expressions
 - elements [405](#)
 - examples [407](#)
 - syntax rules [406](#)
 - groups [403](#)
 - literal values [406](#)
- Properties [20](#)
- Properties view [3](#), [180](#), [402](#)

Q

- QMF Calculator view [3](#)
- QMF Catalog folder [19](#)
- QMF catalog objects [52](#)
- QMF for Windows Configuration Settings [45](#)
- QMF perspective [3](#)
- queries
 - analytical [55](#), [56](#)
 - editors [56](#)
 - OLAP

- queries (*continued*)
 - OLAP (*continued*)
 - building offline [88](#)
 - building online [89](#)
 - filtering cube data [90](#)
 - modifying [88](#)
 - opening [88](#)
 - retrieving cube data [87](#)
 - profiling [115](#)
 - relational [55](#)
 - specifying in visual dashboards [396](#)
 - specifying in visual reports [396](#)
 - standard [55](#)
 - types [55](#)
 - visual [55](#), [56](#), [58](#)
- query [422](#)
- Query Builder
 - add columns [64](#), [71](#)
- Query Builder editor
 - add tables [68](#)
 - fields [67](#), [70](#)
 - global row conditions [67](#), [72](#)
 - join conditions [69](#)
 - SQL Editor [67](#), [74](#)
 - tables [67](#)
- Query Editor [117](#)
- query environment [533](#)
- query menu
 - set row limit [179](#)
- query parameters [483](#)
- query results
 - calculated columns [190](#)
 - categorization
 - adding conformance rules [194](#)
 - adding expressions [194](#)
 - contents of a cell [213](#)
 - email [197](#)
 - exporting [196](#)
 - exporting to a database [198](#)
 - exporting to a file [200](#)
 - exporting to Excel [200](#)
 - file export options [201](#)
 - filtering
 - compound condition [188](#)
 - free style condition [189](#)
 - formatting [180](#)
 - grouping and aggregation [195](#)
 - LOB data [211](#)
 - settings [179](#)
 - sorting [216](#)
- Query Structure tree [87](#), [88](#)
- QueryName [141](#)
- quick reports
 - changing data source [283](#)
 - column headers [270](#)
 - creat [268](#), [272](#)
 - creating [265](#)
 - creating column groups [271](#)
 - data format options [280](#)
 - detail columns [265](#)
 - details columns formatting [278](#)
 - displaying details columns [265](#)
 - displaying grouping columns [265](#)
 - editing [282](#)

- quick reports (*continued*)
 - editing data source [283](#)
 - grouping columns
 - footer [276](#)
 - header [276](#)
 - grouping columns formatting [275](#)
 - individual column formatting [279](#)
 - modifying [284](#)
 - report settings formatting [267](#)
 - working with [264](#)

R

- range chart format options [320](#)
- Recently Used folder [26](#)
- REFn [252](#)
- Refresh [20](#)
- Refresh Component [382](#)
- regular filter [90](#)
- regular save [218](#)
- relational queries [55](#)
- Reload scene [378](#)
- remote server [12](#)
- Remove from Favorites [27](#)
- Remove from Startup [27](#)
- Rename [20](#)
- reports
 - classic
 - data source [246](#)
 - designing forms [248](#)
 - forms [248](#)
 - saving [263](#)
 - visual [335](#)
- repositories
 - connections
 - import [34](#)
 - creating connections
 - file-based [38](#)
 - network [38](#)
 - shared [36](#)
 - web service [40](#)
 - creating new objects [46](#)
 - creating web links
 - data objects [48](#)
 - editing connections [41](#)
 - managing connections [41](#)
 - personal [30](#)
 - personal repositories [30](#)
 - removing connections [41](#)
- repository connection settings [1](#)
- repository connections [30](#), [34](#)
- Repository Connections view [3](#)
- repository objects
 - folders [46](#)
 - forms [46](#)
 - links [46](#)
 - OLAP queries [46](#)
 - personal repositories [46](#)
 - procedures [46](#)
 - queries [46](#)
 - queries using the Draw Query wizard [46](#)
 - repository connections [46](#)
 - visual dashboard template categories [46](#)
 - visual dashboards [46](#)

- repository objects (*continued*)
 - visual report template categories [46](#)
 - visual reports [46](#)
 - workspaces [46](#)
- repository objects caching [42](#)
- Reset All [214](#)
- Reset Formatting [214](#)
- RESET GLOBAL
 - VariableName [158](#)
- resetting perspectives [8](#)
- resource limits
 - binding [541](#)
 - cache [544](#)
 - limits [538](#)
 - LOB [212](#)
 - LOB options [543](#)
 - object list [544](#)
 - object tracking [542](#)
 - options [539](#)
 - save data [541](#)
 - SQL verbs [538](#)
 - timeouts [537](#)
 - viewing [536](#)
- resource manipulation
 - copy [234](#)
 - send mail [235](#)
- Result set pane [323](#), [324](#)
- Results menu commands
 - Add Calculated Column [214](#)
 - AutoFit [214](#)
 - Display Excel Sheet [214](#)
 - Display Report [214](#)
 - Drill Down [214](#)
 - Drill Up [214](#)
 - Export [214](#)
 - Find [214](#)
 - Font [214](#)
 - Format [214](#)
 - Go to [214](#)
 - Grouping and Aggregation [214](#)
 - Reset All [214](#)
 - Reset Formatting [214](#)
 - Retrieve All [214](#)
 - Retrieve All LOB Values [214](#)
 - Show Summaries [214](#)
 - Show Total Summaries [214](#)
 - Sort [214](#)
- results page [87](#)
- Results tab [88–90](#)
- RESULTSET [169](#)
- Retrieve All [214](#)
- Retrieve All LOB Values [214](#)
- REXX Console view [133](#)
- right outer join [63](#), [69](#)
- RIGHTMARGIN [155](#)
- ROWIDADD [164](#)
- ROWIDDISP [164](#)
- ROWIDNAME [164](#)
- ROWLIMIT [158](#)
- Run [20](#)
- RUN
 - &&Variable [158](#)
 - CONFIRM [158](#)
 - FORM [158](#)

RUN (*continued*)
 ObjectName [158](#)
 ROWLIMIT [158](#)
Run Query [287](#)
Runtime mode [401](#)

S

save [49](#)
SAVE
 ACTION [164](#)
 COMMENT [164](#)
 CONFIRM [164](#)
 FOLDER [164](#)
 LANGUAGE [164](#)
 METHOD [164](#)
 MODE [164](#)
 ObjectName [164](#)
 ROWIDADD [164](#)
 ROWIDDISP [164](#)
 ROWIDDISPnnn [164](#)
 ROWIDNAME [164](#)
 SCOPE [164](#)
 SHARE [164](#)
 SPACE [164](#)
save all [49](#)
save at [49](#)
Save Cookie [391](#)
save data objects as dynamarts [210](#)
save data resource limits [541](#)
save methods
 fast [218](#)
 regular [218](#)
scene parameters [483](#)
scene views [356](#)
SceneCenter [361](#)
scenes [354](#)
schedule types
 always expires [344](#)
 at specific date [344](#)
 daily [344](#)
 monthly [344](#)
 never expires [344](#)
 weekly [344](#)
scheduled tasks
 action groups [236–240](#)
 actions [233, 236–240](#)
 resource manipulation [233](#)
 scheduling [241](#)
SCOPE [164](#)
Search configuration [15, 21](#)
Search view [3](#)
security lists [477, 481](#)
Select [408](#)
Send Email [371](#)
Send Event to Embedded Scene [371](#)
SEPARATOR [145](#)
series chart format options [315](#)
Server-Side File System
 preferences [17](#)
ServerName [141](#)
service information [xiii](#)
Set Data Source [284](#)
Set focus [378](#)

Set Font [284](#)
SET GLOBAL
 Value [166](#)
 VariableName [166](#)
SET INVISIBLE
 Value [167](#)
 VariableName [167](#)
SET LOCAL
 Value [167](#)
 VariableName [167](#)
SET LOCAL WITH VALUES
 Value [168](#)
 VariableName [168](#)
Set Name Filter [20](#)
SET OPTIONS
 STOPONERROR [168](#)
 SUPPRESSMESSAGES [168](#)
set resource limit [179](#)
set row limit [179](#)
Set User Information [20, 284](#)
Set values [363, 366, 413](#)
SHARE [150, 164](#)
shared prompt [239, 240](#)
shared repositories [35, 36, 42](#)
SHOW
 RESULTSET [169](#)
 VIEW [169](#)
Show dialog [384](#)
Show Forecast Outline Tree [287](#)
Show in Repository [20](#)
Show message [363, 375, 413](#)
Show References [400](#)
Show Summaries [214](#)
Show Total Summaries [214](#)
show view [7](#)
Side Group [214](#)
Side groups [325](#)
Side Groups [324](#)
silent installation [549](#)
simple moving average [297](#)
simple prompts
 adding [78](#)
 working with [77](#)
single exponential smoothing [297](#)
Single Exponential Smoothing [287](#)
SIZE [155](#)
SMTP preferences [16](#)
SMTPPASSWORD [152](#)
SMTPSERVER [152](#)
SMTPUSER [152](#)
Solaris [1](#)
Sort [214](#)
SPACE [150, 158, 164](#)
SPACE DATABASE [150, 158, 164](#)
special registers [124](#)
SPLIT [145](#)
SQL
 fonts [129](#)
SQL page [87](#)
SQL prompts [76](#)
SQL Query editor
 content assist [59](#)
 parameter hints [60](#)
SQL Server [1](#)

- SQL verbs resource limits [538](#)
- Standard Deviation [214](#)
- Startup folder [26](#)
- static SQL [1](#)
- static SQL packages [36](#)
- STDEV [261](#)
- stock images
 - creating [482](#)
 - using [483](#)
- stored procedures [36](#)
- strategy options
 - aggregation strategy [294](#)
 - construction strategy [294](#)
 - distribution strategy [294](#)
 - forecast column [294](#)
- SUBJECT [152](#)
- substitution variables [131](#)
- Sum [214](#)
- SUM [261](#)
- support information [xiii](#)
- syntax diagrams [138](#)
- syntax diagrams, how to read [xiii](#)

T

- tab group [7](#)
- Table Editor
 - Prompted tab [121](#), [123](#)
 - row conditions [123](#), [124](#)
 - sort conditions [121](#), [122](#)
 - SQL tab [122](#), [124](#)
- Table Viewer [117](#)
- TCPCT [261](#)
- template categories
 - creating [336](#)
- templates [335](#)
- Templates
 - visual data objects [335](#)
- Text [355](#)
- text fields [252](#)
- TextSet [355](#)
- Third Party Libraries preferences [18](#)
- time data edit codes [521](#)
- time-series data [287](#)
- timeouts resource limits [537](#)
- Timer object [413](#)
- timestamp data edit codes [521](#)
- title chart format options [320](#)
- Top Group [214](#)
- Top groups [325](#)
- Top Groups [324](#)
- TOPMARGIN [155](#)
- TPCT [261](#)
- tracking signal [301](#)
- translation tables
 - switching [490](#)
- TYPE [152](#)
- Type of sort
 - ascending [216](#)
 - descending [216](#)
- TYPEFACE [155](#)

U

- UNICODE [145](#)
- union queries [97](#)
- UNITS [145](#)
- URL template [13](#), [33](#)
- usage codes
 - ACROSS [261](#)
 - AVERAGE [261](#)
 - Breakn [261](#)
 - BREAKnX [261](#)
 - CALCid [261](#)
 - COUNT [261](#)
 - CPCT [261](#)
 - CSUM [261](#)
 - FIRST [261](#)
 - GROUP [261](#)
 - LAST [261](#)
 - MAXIMUM [261](#)
 - MINIMUM [261](#)
 - OMIT [261](#)
 - PCT [261](#)
 - STDEV [261](#)
 - SUM [261](#)
 - TCPCT [261](#)
 - TPCT [261](#)
- USE REPOSITORY
 - RepositoryName [170](#)
- USEFORMPS [155](#)
- user defined edit codes [522](#)
- user information [31](#)
- user interface [31](#)
- user menus
 - Form [284](#)
 - results [214](#)
- User perspective [1](#), [3](#), [6](#)
- user-defined functions [191](#), [192](#), [394](#)
- UserName [141](#)

V

- Value [151](#), [166–168](#)
- value(Y)series chart format options [316](#)
- Values [325](#)
- VariableName [151](#), [158](#), [166–168](#)
- Variables bar [404](#)
- view
 - resource limit [179](#)
- VIEW [169](#)
- viewpoints
 - modifying [360](#)
 - navigation [359](#)
- views
 - Analyzer [491](#)
 - Cheat Sheets [3](#)
 - Data Source Connections [3](#)
 - Events [3](#), [407](#)
 - Expressions [493](#)
 - Help [3](#)
 - Layers [3](#), [477](#)
 - Layout [89](#)
 - opening [7](#)
 - Outline [3](#)
 - Output [3](#)

- views (*continued*)
 - Palette [3](#), [399](#), [408](#)
 - Personal [3](#), [26](#)
 - positioning [7](#)
 - Profiler [492](#)
 - Progress [3](#)
 - Project Explorer [3](#), [337](#), [399](#), [400](#)
 - Properties [3](#), [180](#), [399](#), [402](#)
 - QMF Calculator [3](#)
 - Repository Connections [3](#)
 - Results [219](#), [311](#)
 - REXX Console [133](#)
 - Search [3](#)
 - Workspaces [3](#), [19](#)
 - views and perspectives [6](#)
 - virtual tables
 - structure [501](#), [502](#)
 - visual applications [335](#)
 - visual dashboard
 - actions
 - Show dialog [389](#)
 - visual dashboard templates [352](#)
 - visual dashboards
 - actions
 - Close dialog [390](#)
 - Create File [380](#)
 - End Session [379](#)
 - Execute JavaScript [370](#)
 - Execute procedure [369](#)
 - Execute shell command [372](#)
 - Execute SQL statement [368](#)
 - Export to excel [376](#)
 - Export to PDF [379](#)
 - Invalidate Query Cache [382](#)
 - Jump to new location [364](#)
 - Launch LOB [383](#)
 - Load Cookie [392](#)
 - Navigate back [367](#)
 - Navigate to an embedded scene [366](#)
 - Open URL [391](#)
 - Print scene [370](#)
 - Refresh Component [382](#)
 - Reload scene [378](#)
 - Save Cookie [391](#)
 - Send Email [371](#)
 - Send Event to Embedded Scene [371](#)
 - Set focus [378](#)
 - Set values [366](#)
 - Show message [375](#)
 - adding [354](#)
 - creating [351](#)
 - creating scene views [356](#)
 - data source connections [353](#), [395](#)
 - deleting [354](#)
 - drilldown [358](#)
 - embedded scenes
 - parameterized [362](#)
 - event actions [363](#)
 - modal dialog [384](#)–[388](#)
 - navigation features [358](#)
 - open queries [494](#)
 - planning [350](#)
 - queries [342](#), [396](#)
 - read-only [395](#)
 - visual dashboards (*continued*)
 - renaming [354](#)
 - scenes
 - parameters [355](#)
 - specifying data source connections [341](#), [395](#)
 - storyboards [350](#)
 - template category [336](#)
 - templates [335](#), [352](#)
 - viewpoint navigation [359](#), [360](#)
 - working in version control systems
 - saving as folders [393](#)
 - visual designer
 - editor controls [401](#)
 - editor window [401](#)
 - grid [401](#)
 - guides [401](#)
 - rulers [401](#)
 - Snap to Grid [401](#)
 - Visual Designer
 - preferences [18](#)
 - Visual Designer perspective [3](#), [338](#)
 - visual projects [400](#)
 - visual report
 - read-only [395](#)
 - visual report main pages
 - Detail [343](#)
 - Detail Footer [343](#)
 - Detail Header [343](#)
 - Page Footer [343](#)
 - Page Header [343](#)
 - Report Footer [343](#)
 - Report Header [343](#)
 - visual reports
 - converting to offline [346](#)
 - creating [338](#)
 - data retrieval [344](#)
 - data source connections [395](#)
 - fixed pages [343](#)
 - legacy
 - importing [340](#)
 - opening [340](#)
 - main pages [343](#)
 - open queries [494](#)
 - queries
 - data retrieval schedules [346](#)
 - specifying data source connections [341](#), [395](#)
 - template category [336](#)
 - templates [335](#), [339](#)
 - visual reports fixed pages
 - Fixed1 [343](#)
 - Fixed2 [343](#)
 - PageFooting [343](#)
 - PageHeading [343](#)
- W**
 - web service repositories [35](#), [40](#)
 - weighted moving average [297](#)
 - Weighted Moving Average [287](#)
 - WIDTH [145](#), [155](#)
 - workspace
 - publishing to QMF Vision [116](#)
 - workspace properties
 - Connection Parameters [24](#)

workspace properties (*continued*)

[LDAP 24](#)

[Login mapping 24](#)

[Plug-ins 24](#)

[Resource Limits 24](#)

[Security 24](#)

workspaces

[adding a new workspace 26](#)

[adding content 25](#)

[creating 26](#)

[discovering contents 19](#)

[filtering data source objects 23](#)

[folders 22](#)

[linking to objects](#)

[repository 23](#)

[workspace 23](#)

[moving objects 25](#)

[properties 24](#)

[refreshing contents 25](#)

[Workspaces view 3, 19](#)

Workspaces view menu commands

[Add to Favorites 20](#)

[Add to Startup 20](#)

[Copy 20](#)

[Delete 20](#)

[Explore 20](#)

[New 20](#)

[Open 20](#)

[Open From 20](#)

[Paste 20](#)

[Paste Link 20](#)

[Properties 20](#)

[Refresh 20](#)

[Rename 20](#)

[Run 20](#)

[Set Name Filter 20](#)

[Set User Information 20](#)

[Show in Repository 20](#)

X

[X-Axis chart format options 318](#)

[Xlocation 359](#)

[XMLA connectivity 1](#)

Y

[Y-Axis chart format options 318](#)

[Ylocation 359](#)

Z

[zoom 213](#)

[ZoomPct 361](#)



Product Number: 5737-E42

SC27-8867

