IBM DataQuant for z/OS and Multiplatforms
Version 1 Release 2

Getting Started with DataQuant

IBM
IBM DataQuant for z/OS and Multiplatforms
Version 1 Release 2

 Getting Started with DataQuant

IBM
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.

This edition applies to Version 1 Release 2 of IBM DataQuant for z/OS, program number 5697-N64. It also applies to program number 5724-R90, which applies to Version 1 Release 2 of the following products: IBM DataQuant for Multiplatforms; IBM DataQuant for DB2 Warehouse Starter Edition; and IBM DataQuant for DB2 Warehouse Intermediate Edition. This information applies to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC19-1165-15.

© Rocket Software Inc. 1995, 2015
# Contents

## Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ix</td>
<td></td>
</tr>
</tbody>
</table>

## About this information

<table>
<thead>
<tr>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who should read this information</td>
<td>xi</td>
</tr>
<tr>
<td>Service updates and support information</td>
<td>xi</td>
</tr>
<tr>
<td>Highlighting conventions</td>
<td>xi</td>
</tr>
<tr>
<td>How to read syntax diagrams</td>
<td>xi</td>
</tr>
<tr>
<td>How to look up message explanations</td>
<td>xiii</td>
</tr>
<tr>
<td>How to send your comments</td>
<td>xiv</td>
</tr>
</tbody>
</table>

## Chapter 1. DataQuant overview

<table>
<thead>
<tr>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>User perspective overview</td>
<td>1</td>
</tr>
<tr>
<td>Administrator perspective overview</td>
<td>2</td>
</tr>
</tbody>
</table>

## Chapter 2. Using perspectives to set up the User and Administrator application interface

<table>
<thead>
<tr>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working with views and perspectives</td>
<td>9</td>
</tr>
<tr>
<td>Opening perspectives</td>
<td>9</td>
</tr>
<tr>
<td>Closing perspectives</td>
<td>10</td>
</tr>
<tr>
<td>Opening views</td>
<td>10</td>
</tr>
<tr>
<td>Positioning views</td>
<td>10</td>
</tr>
<tr>
<td>Configuring perspectives</td>
<td>10</td>
</tr>
<tr>
<td>Resetting perspectives</td>
<td>11</td>
</tr>
<tr>
<td>Setting preferences for users</td>
<td>11</td>
</tr>
<tr>
<td>Setting general preferences</td>
<td>12</td>
</tr>
<tr>
<td>Setting appearance preferences</td>
<td>13</td>
</tr>
<tr>
<td>Setting dashboard runtime preferences</td>
<td>13</td>
</tr>
<tr>
<td>Setting data management preferences</td>
<td>14</td>
</tr>
<tr>
<td>Setting help preferences</td>
<td>14</td>
</tr>
<tr>
<td>Setting JDBC preferences</td>
<td>15</td>
</tr>
<tr>
<td>Setting LOB preferences</td>
<td>17</td>
</tr>
<tr>
<td>Setting log preferences</td>
<td>18</td>
</tr>
<tr>
<td>Setting the directory location preference for objects generated by Export actions</td>
<td>18</td>
</tr>
<tr>
<td>Setting report design preferences</td>
<td>19</td>
</tr>
<tr>
<td>Setting themes preferences</td>
<td>19</td>
</tr>
<tr>
<td>Setting third party library preferences</td>
<td>20</td>
</tr>
<tr>
<td>Specifying Visual Designer preferences</td>
<td>20</td>
</tr>
<tr>
<td>Exploring the Workspaces view</td>
<td>21</td>
</tr>
<tr>
<td>Discovering the contents of a workspace</td>
<td>22</td>
</tr>
<tr>
<td>Workspace menu commands</td>
<td>22</td>
</tr>
<tr>
<td>Managing workspace folders</td>
<td>24</td>
</tr>
<tr>
<td>Filtering data source objects</td>
<td>24</td>
</tr>
<tr>
<td>Creating links to workspace objects</td>
<td>24</td>
</tr>
<tr>
<td>Creating links to repository objects</td>
<td>25</td>
</tr>
<tr>
<td>Viewing properties for workspace objects</td>
<td>25</td>
</tr>
<tr>
<td>Refreshing workspace contents</td>
<td>26</td>
</tr>
<tr>
<td>Moving workspace objects</td>
<td>26</td>
</tr>
<tr>
<td>Adding content to a workspace</td>
<td>26</td>
</tr>
<tr>
<td>Creating additional workspaces</td>
<td>27</td>
</tr>
<tr>
<td>Working in the Personal view</td>
<td>28</td>
</tr>
<tr>
<td>Personal view menu commands</td>
<td>28</td>
</tr>
<tr>
<td>Working with database tables</td>
<td>28</td>
</tr>
<tr>
<td>Specifying sort conditions for tables</td>
<td>30</td>
</tr>
<tr>
<td>Specifying row conditions for tables</td>
<td>31</td>
</tr>
<tr>
<td>Editing database tables using DB2 special registers</td>
<td>32</td>
</tr>
<tr>
<td>Creating personal repositories</td>
<td>33</td>
</tr>
</tbody>
</table>
Connecting to a repository ........................................... 34
Specifying user information ......................................... 34
Changing user interface configuration information ............ 35
  Setting up JDBC driver information .......................... 35
  Setting up repository connection information ............ 37
Changing your DB2 password .......................................... 46
Configuring repository caching ........................................ 46

Chapter 3. Creating new objects or repository items ............... 51
Correct format for identifiers ......................................... 52
Matching patterns ....................................................... 52
Creating queries .......................................................... 53
  Creating queries using SQL ...................................... 53
  Creating visual queries using SQL ............................. 54
Using Content Assist for queries and visual queries .......... 55
Using Parameter Hints for queries and visual queries ....... 56
Creating queries using the Prompted Query editor ............ 56
Creating queries using the Diagram Query editor ............. 64
Creating visual queries using the Diagram Query editor .... 66
Adding tables ............................................................. 68
Editing join conditions for multiple tables ...................... 69
Creating queries using the Draw Query wizard ................. 70
Working with prompts in queries .................................. 71
Specifying substitution variables ................................... 77
Setting fonts for SQL statements ................................... 78
Query menu ............................................................... 78
Creating OLAP queries ................................................ 79
  Retrieving cube data .............................................. 79
  Opening existing OLAP queries ................................. 80
Modifying OLAP queries to obtain different views of cube data 80
Working with OLAP query results ................................... 83
  OLAP Query menu ................................................... 83
Creating analytical queries ............................................ 84
  Adding an append query .......................................... 85
  Adding a join query .............................................. 86
  Adding a crosstab query ........................................ 86
  Adding an analytical query ..................................... 87
  Adding a normalize query ...................................... 87
  Adding a conditional grouping .................................. 90
Saving objects ............................................................ 92
  Saving objects in a repository or a repository workspace 92
  Saving objects to a file ......................................... 93
  Saving objects in the QMF catalog ............................ 93
Opening saved objects ................................................ 94
  Opening repository objects ..................................... 94
  Opening objects saved to a file ................................ 94
  Opening QMF catalog objects ................................... 94

Chapter 4. Managing query results .................................. 97
Formatting the query results .......................................... 97
  Using the Properties view ....................................... 97
  Using the Layout Properties dialog box ...................... 97
Filtering query results .................................................. 104
  Adding calculated columns to the query results .......... 106
Applying grouping and aggregation to query result columns 107
  Grouping and aggregation fields ................................ 108
Exporting query results ............................................... 109
  Sending query results in an email ............................ 110
  Exporting query results to a database ...................... 111
  Exporting query results to a file ............................. 112
Importing query results from CSV and text files ........................................... 115
   Specifying separator options for CSV and TXT files ..................................... 115
   Specifying the result set structure of CSV and Text files ............................... 116
Importing data contained in IXF files .......................................................... 117
   Opening the contents of an IXF file for further processing ............................ 117
   Directly importing an IXF file into a database table ..................................... 118
Importing table contents of a saved file into a database table ......................... 118
Working with LOB data in the query results ................................................. 119
   Overriding LOB resource limit option ......................................................... 121
Viewing and editing the contents of a cell in the query results ....................... 121
Viewing multiple result sets ............................................................................. 122
Results menu ..................................................................................................... 122
Sorting query results ......................................................................................... 124
Saving data using the regular and fast save methods ....................................... 125

Chapter 5. Generating reports ........................................................................ 127
Creating a new report ......................................................................................... 127
Selecting a classic report’s data source object ............................................... 128
   Listing data source objects .............................................................................. 129
Exporting classic and visual reports to a file ................................................... 129
Creating classic report forms ........................................................................... 130
   Designing a classic report form ....................................................................... 130
   Using form variables ....................................................................................... 143
   Using usage codes. ........................................................................................... 144
Saving classic reports. ...................................................................................... 146
   Saving a form with an associated data source object ...................................... 147
   Saving a form without an associated data source object ............................... 147
Working with BIRT reports .............................................................................. 147
Form menu ......................................................................................................... 149

Chapter 6. Working with Forecasts ................................................................ 151
Creating Forecasts ............................................................................................. 152
Specifying forecast data source options ........................................................... 153
Specifying forecast date parameters ................................................................... 153
Specifying forecast grouping hierarchy options .............................................. 154
Specifying forecast strategy options ................................................................... 154
Specifying forecasting models options ............................................................ 155
   Setting simple moving average parameters .................................................... 156
   Setting weighted moving average parameters .............................................. 156
   Setting single exponential smoothing parameters ....................................... 156
   Setting double exponential smoothing parameters ...................................... 157
   Setting Holt-Winters method parameters ...................................................... 157
   Setting multiplicative decomposition parameters ........................................ 158
   Setting curve fitting parameters ................................................................... 158
   Setting polynomial regression parameters ................................................... 158
   Setting neural network parameters ............................................................... 159
Monitoring forecast model performance ............................................................ 159
Observing trends ................................................................................................ 160
Observing seasonality ......................................................................................... 161
Observing cyclicity .............................................................................................. 161
Editing chart properties ...................................................................................... 161
Editing grid properties ......................................................................................... 162

Chapter 7. Working with charts. .................................................................... 163
Creating a chart from the query results grid ..................................................... 164
Creating a chart from the Display editor ............................................................ 164
Editing query result set columns ....................................................................... 165
Editing display mode layout structures .............................................................. 165
Managing display modes .................................................................................... 166
   Adding a display mode ..................................................................................... 166
Chapter 8. Working with visual projects .............................................. 175
Creating visual reports ........................................................................ 175
Creating reports using the visual report wizard .................................... 177
Opening a legacy visual report ............................................................... 178
Specifying data source connections for visual reports ......................... 180
Specifying visual report queries ........................................................... 180
Designing the main page of a visual report .......................................... 181
Designing a fixed page for a visual report ............................................. 182
Managing report data retrieval ............................................................... 183
Creating Visual dashboards. ................................................................. 185
Planning your dashboard ..................................................................... 187
Creating Visual Dashboards using the Visual Dashboard wizard ............ 188
Importing a legacy Visionary world ...................................................... 190
Specifying data source connections for visual dashboards .................. 191
Specifying visual dashboard queries .................................................... 192
Working with scenes ......................................................................... 193
Protecting visual dashboard data ......................................................... 222
Queries and visual projects ................................................................. 222
Specifying visual report queries ......................................................... 223
Working with the Visual Designer ........................................................ 223
Working with the Project Explorer ....................................................... 224
Working with the editor ..................................................................... 225
Rulers, guides, and the grid ................................................................. 226
Working with the Properties view ........................................................ 227
Working with the Events view ............................................................... 231
Working with the Palette view ............................................................. 232
Working with the Layers view .............................................................. 271
Working with global resources ............................................................. 271

Chapter 9. Working with procedures .................................................. 283
Creating a new procedure ................................................................... 283
Scheduling procedures to run ............................................................... 284
Viewing scheduled jobs ..................................................................... 285
Restoring scheduled jobs after logon information change ..................... 286
Converting legacy job definition files .................................................. 287
Procedure commands. ...................................................................... 287
Interface differences ........................................................................ 287
Procedure syntax ............................................................................. 288
Reading the syntax diagrams ............................................................... 288
QMF Catalog objects versus Repository folder objects ........................ 289
BOTTOM command ......................................................................... 291
CONNECT command ........................................................................ 291
CONVERT command. ...................................................................... 291
DISPLAY command ......................................................................... 292
DRAW command ............................................................................... 294
EDIT command ................................................................................ 294
ERASE command ............................................................................ 295
EXECUTE command ......................................................................... 295
EXPORT command ........................................................................... 296
FORWARD command ......................................................................... 300
IMPORT command ............................................................................ 300
LIMIT LOCAL command .................................................................. 301
MAIL TO Command ......................................................................... 301
PRINT command ............................................................................... 305
RESET GLOBAL ............................................................................... 307
RUN command. ................................................................................ 307
Appendix C. Messages ........................................... 361

Notices ............................................................. 425
  Trademarks ..................................................... 427
  Terms and conditions for product documentation .......... 427
  Privacy policy considerations .................................. 428

Glossary .......................................................... 429

Index ............................................................... 435
## Tables

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>User perspective: Views</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>Administrator perspective: Views</td>
<td>6</td>
</tr>
<tr>
<td>3.</td>
<td>Visual Designer perspective: Views</td>
<td>8</td>
</tr>
<tr>
<td>4.</td>
<td>Report Design perspective: Views</td>
<td>8</td>
</tr>
<tr>
<td>5.</td>
<td>Prompt types</td>
<td>71</td>
</tr>
<tr>
<td>6.</td>
<td>Original result set data</td>
<td>88</td>
</tr>
<tr>
<td>7.</td>
<td>Normalized query results</td>
<td>88</td>
</tr>
<tr>
<td>8.</td>
<td>Alternate normalized query results</td>
<td>88</td>
</tr>
<tr>
<td>9.</td>
<td>Crosstabbed query results for furniture sales</td>
<td>89</td>
</tr>
<tr>
<td>10.</td>
<td>Normalized query results for furniture sales</td>
<td>89</td>
</tr>
<tr>
<td>11.</td>
<td>Original data</td>
<td>90</td>
</tr>
<tr>
<td>12.</td>
<td>Conditionally grouped data</td>
<td>90</td>
</tr>
<tr>
<td>13.</td>
<td>Text options</td>
<td>113</td>
</tr>
<tr>
<td>14.</td>
<td>Main section elements for a classic report</td>
<td>131</td>
</tr>
<tr>
<td>15.</td>
<td>Breaks section elements for a classic report</td>
<td>133</td>
</tr>
<tr>
<td>16.</td>
<td>HTML variables</td>
<td>135</td>
</tr>
<tr>
<td>17.</td>
<td>Calculations option elements for a classic report</td>
<td>136</td>
</tr>
<tr>
<td>18.</td>
<td>Columns option elements for a classic report</td>
<td>137</td>
</tr>
<tr>
<td>19.</td>
<td>Conditions option elements for a classic report</td>
<td>138</td>
</tr>
<tr>
<td>20.</td>
<td>Detail section elements for a classic report</td>
<td>138</td>
</tr>
<tr>
<td>21.</td>
<td>Options section elements for a classic report</td>
<td>140</td>
</tr>
<tr>
<td>22.</td>
<td>Final section elements for a classic report</td>
<td>141</td>
</tr>
<tr>
<td>23.</td>
<td>Page section elements for a classic report</td>
<td>142</td>
</tr>
<tr>
<td>24.</td>
<td>Command line parameters</td>
<td>340</td>
</tr>
</tbody>
</table>
About this information

This information describes how to use the DataQuant for Workstation and DataQuant 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 DataQuant for Workstation and DataQuant 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:


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).

---

1. Throughout this information, the IBM® DataQuant client for WebSphere Application Server is referred to as DataQuant for WebSphere and the IBM DataQuant client for the workstation environment is referred to as DataQuant for Workstation.
• Optional items appear below the main path.

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.

• 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

In the search box that is in the upper left toolbar of any Eclipse help system, such as the IBM Information Management Software for z/OS Solutions Information Center enter the number of the message that you want to locate. For example, you can enter DFS1065A in the search field.

Use the following tips to help you improve your message searches:
• You can search for information about codes by entering the code; for example, enter -327.
• Enter the complete or partial message number. You can use wildcards 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 number and type of characters, 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 single character or values.

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 are 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/OS® elements and features, z/VM®, VSE/ESA, and Clusters for AIX® and Linux:
• The Internet. You can access IBM message explanations directly from the LookAt website at [http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/](http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/)

• Your z/OS TSO/e host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/e command line (for example, a TSO/e prompt, ISPF, or z/OS UNIX System Services running OMVS).

• Your Microsoft Windows workstation. You can install code to access IBM message explanations on the z/OS Collection (SK3T-4271) using LookAt from a Microsoft Windows command prompt (also known as the DOS command line).

• 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 website.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from a disk on your z/OS Collection (SK3T-4271) or from the LookAt website (click [Download](http://www.ibm.com/software/data/rcf) and select the platform, release, collection, and location that suit your needs). 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 documentation, use either of the following options:

- Use the online reader comment form, which is located at: [http://www.ibm.com/software/data/rcf](http://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 your product, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).
Chapter 1. DataQuant overview

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

DataQuant is an Eclipse-based Business Intelligence tool that transforms business data into a visual information platform with visual data on-demand. Using the DataQuant 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 end user and database platforms.
- Supports DB2 V9 for z/OS support for SQL stored procedures.
- Supports DB2 V10 for z/OS functionality, including DB2 V10 data types.
- 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.
- DataQuant for Workstation is an Eclipse-based, cross-platform, rich workstation application that can be run on Microsoft® Windows®, Linux®, and Macintosh®.
- DataQuant 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.

To get started using DataQuant, 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.

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
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 DataQuant 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 repository workspaces. The data sources and objects that you can access from the workspace are displayed in the Workspaces view.
Chapter 2. Using perspectives to set up the User and Administrator application interface

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

**Perspectives**

DataQuant utilizes and employs the Eclipse-based concept of Perspectives, Views, and Editors.

The sections that follow explain Perspectives, Views, and Editors as used by DataQuant. 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 DataQuant.

The Perspectives that come with DataQuant are configured with a 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 DataQuant, it opens to a specific perspective and set of views.

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

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

- **User** which provides the views, menus, and wizards that enable you to perform an extensive range of query and reporting functions. The User perspective defines the initial set and layout of the views that you will use to query data sources and work with query results.
- **Administrator** which provides the views, menus, and wizards that enable a user with administrative privileges to create and maintain repositories. Not all instances of DataQuant are installed with the Administrator perspective.
- **QMF** which provides the views, menus, and wizards that are most appropriate for working with QMF™ catalogs.
- **Visual Designer** which provides the views, menus, and wizards that you will use to create visual reports and dashboards. (DataQuant for Workstation only)
- **Report Design** which provides the views, menus, and wizards that you will use to create BIRT reports. (DataQuant for Workstation only)
Views

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

DataQuant 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, Report Design, QMF, and Administrator) have a designated or predetermined set of Views. Each view displays as a pane in the main workstation window. Each perspective of DataQuant initially displays the views that are most commonly used to perform specific functions.

Available views in the User perspective

Users and administrators use views in the User perspective to manage the query, reporting and visual project development environment.

The following table lists the commonly used views (Initial default views), and other views that are available from the User perspective.

<table>
<thead>
<tr>
<th>View Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial default views</strong></td>
<td></td>
</tr>
<tr>
<td>Personal</td>
<td>The Personal view maintains and displays objects in the Recently Used, Favorites, and Startup folders.</td>
</tr>
<tr>
<td>Properties</td>
<td>The Properties view displays the properties of the object that is selected or active in the editor or another view.</td>
</tr>
<tr>
<td>Outline</td>
<td>The Outline view displays an outline of the active object in the editor. The structural elements that are listed depend on the type of object that is open.</td>
</tr>
<tr>
<td>Workspaces</td>
<td>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.</td>
</tr>
<tr>
<td>Repository Connections</td>
<td>The Repository Connections view displays the list of repository connections available to you.</td>
</tr>
</tbody>
</table>

**Additional views you might use from this perspective**

<table>
<thead>
<tr>
<th>View Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>The Output view displays informational and error messages generated while running the application interface.</td>
</tr>
<tr>
<td>Palette</td>
<td>The Palette view displays the objects that you can insert in a visual project. (DataQuant for Workstation only)</td>
</tr>
<tr>
<td>Progress</td>
<td>The Progress view displays all progress, percentage of work completed, and cancellation messages.</td>
</tr>
</tbody>
</table>
| Help      | The Help provides access to the following view options:  
• Help option to launch the context Help for the workstation  
• Cheat Sheets view that displays a list of available cheat sheets. |
Table 1. User perspective: Views (continued)

<table>
<thead>
<tr>
<th>View Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataQuant Workbench</td>
<td>The DataQuant Workbench provides access to the following view options:</td>
</tr>
<tr>
<td></td>
<td>• Data Source Connections</td>
</tr>
<tr>
<td></td>
<td>The Data Source Connections view which displays a list of all connections that are currently active or have been made to a data source</td>
</tr>
<tr>
<td></td>
<td>• Personal</td>
</tr>
<tr>
<td></td>
<td>This 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.</td>
</tr>
<tr>
<td></td>
<td>• REXX console</td>
</tr>
<tr>
<td></td>
<td>If you are running a procedure with logic, use the REXX console view to communicate with the procedure.</td>
</tr>
<tr>
<td></td>
<td>• Search</td>
</tr>
<tr>
<td></td>
<td>Use this view search for information within the workstation application.</td>
</tr>
<tr>
<td></td>
<td>• Users and Groups</td>
</tr>
<tr>
<td></td>
<td>This view displays the users and groups created for your internal security provider. This view is only available if you have the Administrator component installed.</td>
</tr>
<tr>
<td></td>
<td>• The Repositories view displays the repositories that you have created. This view is only available if you have the Administrative component installed.</td>
</tr>
<tr>
<td>Visual Designer Category</td>
<td>The DataQuant Workbench view provides the following workbench view options:</td>
</tr>
<tr>
<td></td>
<td>• Events</td>
</tr>
<tr>
<td></td>
<td>Displays navigation elements that you can apply to layout objects within a visual dashboard. (DataQuant for Workstation only)</td>
</tr>
<tr>
<td></td>
<td>• Project Explorer</td>
</tr>
<tr>
<td></td>
<td>Displays each element in a visual report or a visual dashboard.</td>
</tr>
<tr>
<td></td>
<td>• DataQuant Calculator</td>
</tr>
<tr>
<td></td>
<td>Used when creating visual projects. (DataQuant for Workstation only)view which is used</td>
</tr>
<tr>
<td></td>
<td>• Layers</td>
</tr>
<tr>
<td></td>
<td>Used to group palette objects into separate layers that can be displayed or hidden during design time.</td>
</tr>
</tbody>
</table>

Available views in the Administrator perspective

Administrators use views in the Administrator perspective to manage, administer and configure repositories, repository connections, data source connections and users and groups.

The following table lists the commonly used views (Initial default views), and other views that are available from the Administrator perspective.


<table>
<thead>
<tr>
<th>View</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial default views</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Repositories</strong></td>
<td>The <strong>Repositories</strong> view displays the repositories that you have created. This view is only available if you have the Administrator component installed.</td>
</tr>
<tr>
<td><strong>Repository Explorer</strong></td>
<td>The <strong>Repository Explorer</strong> 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.</td>
</tr>
<tr>
<td><strong>Properties</strong></td>
<td>The <strong>Properties</strong> view displays the properties of the object that is selected or active in the editor or another view.</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>The <strong>Output</strong> view displays informational and error messages generated while running the application interface.</td>
</tr>
</tbody>
</table>

**Additional views you might use from this perspective**

<table>
<thead>
<tr>
<th>View</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outline</strong></td>
<td>The <strong>Outline</strong> view displays an outline of the active object in the editor. The structural elements that are listed depend on the type of object that is open.</td>
</tr>
<tr>
<td><strong>Palette</strong></td>
<td>The <strong>Palette</strong> view displays the objects that you can insert in a visual project. (DataQuant for Workstation only)</td>
</tr>
<tr>
<td><strong>Progress</strong></td>
<td>The <strong>Progress</strong> view displays all progress, percentage of work completed and cancellation messages.</td>
</tr>
</tbody>
</table>
| **Help** | The **Help** provides access to the following view options:  
  * The **Help** option that launches the context **Help** for the workstation.  
  * The **Cheat Sheets** view that displays a list of available cheat sheets. (DataQuant for Workstation only) |
Table 2. Administrator perspective: Views (continued)

<table>
<thead>
<tr>
<th>View</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataQuant Workbench</td>
<td>The <strong>DataQuant Workbench</strong> provides access to the following view options:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Data Source Connections</strong></td>
</tr>
<tr>
<td></td>
<td>The <strong>Data Source Connections</strong> view which displays a list of all connections that are currently active or have been made to a data source.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Personal</strong></td>
</tr>
<tr>
<td></td>
<td>This 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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>REXX console</strong></td>
</tr>
<tr>
<td></td>
<td>If you are running a procedure with logic, use the REXX console view to communicate with the procedure.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Search</strong></td>
</tr>
<tr>
<td></td>
<td>Use this view search for information within the workstation application.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Users and Groups</strong></td>
</tr>
<tr>
<td></td>
<td>This view displays the users and groups created for your internal security provider. This view is only available if you have the Administrator perspective installed.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Workspaces</strong></td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Visual Designer</td>
<td>The <strong>DataQuant Workbench</strong> view provides the following workbench view options:</td>
</tr>
<tr>
<td>Category</td>
<td>• <strong>Events</strong></td>
</tr>
<tr>
<td></td>
<td>Displays navigation elements that you can apply to layout objects within a visual dashboard. (DataQuant for Workstation only)</td>
</tr>
<tr>
<td></td>
<td>• <strong>Project Explorer</strong></td>
</tr>
<tr>
<td></td>
<td>Displays each element in a visual report or a visual dashboard.</td>
</tr>
<tr>
<td></td>
<td>• <strong>DataQuant Calculator</strong> view which is used when creating visual projects. (DataQuant for Workstation only)</td>
</tr>
<tr>
<td></td>
<td>• <strong>Layers</strong></td>
</tr>
<tr>
<td></td>
<td>Used to group palette objects into separate layers that can be displayed or hidden during design time.</td>
</tr>
</tbody>
</table>

**Note:** Only one user can be logged in to the Administrator perspective at a time on a given server. If a user attempts to log on to the Administrator perspective while another user is already logged on, an error message opens. The user can either return to the user perspective or quit the current session. If the user quits the current session, all unsaved work is lost.

The following table lists the commonly used views (*Initial default views*), and other views that are available from the DataQuant Classic perspective.

Available views in the Visual Designer perspective
The **Visual Designer** perspective is only available in DataQuant for Workstation.

Users use views in the **Visual Designer** perspective to display elements in visual reports or visual dashboards and the navigation event elements.

The following table lists the *initial default views* from the **Visual Designer** perspective.

### Table 3. Visual Designer perspective: Views

<table>
<thead>
<tr>
<th>View</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial default views</strong></td>
<td></td>
</tr>
<tr>
<td>Workspaces</td>
<td>The <strong>Workspaces</strong> 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</td>
</tr>
<tr>
<td>Project Explorer</td>
<td>The <strong>Project Explorer</strong> view displays each element in a visual report or a visual dashboard.</td>
</tr>
<tr>
<td>Output</td>
<td>The <strong>Output</strong> view displays informational and error messages generated while running the application interface.</td>
</tr>
<tr>
<td>Properties</td>
<td>The <strong>Properties</strong> view displays the properties of the object that is selected or active in the editor or another view</td>
</tr>
<tr>
<td>Events</td>
<td>The <strong>Events</strong> view displays navigation elements that you can apply to layout objects within a visual dashboard. (DataQuant for Workstation only)</td>
</tr>
<tr>
<td>Layers</td>
<td>The <strong>Layers</strong> view groups palette objects into separate layers that can be displayed or hidden during design time.</td>
</tr>
</tbody>
</table>

**Available views in the Report Design perspective**

Users use views in the **Report Design** perspective to display existing artifacts for use when designing reports, as well as BIRT report elements.

The following table lists the *initial default views* from the **Report Design** perspective.

### Table 4. Report Design perspective: Views

<table>
<thead>
<tr>
<th>View</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Explorer</strong></td>
<td>The <strong>Data Explorer</strong> view displays the data source and data set information for a BIRT report.</td>
</tr>
<tr>
<td><strong>Resource Explorer</strong></td>
<td>The <strong>Resource Explorer</strong> view displays the objects that have been created for reuse in a BIRT report.</td>
</tr>
<tr>
<td><strong>Property Editor</strong></td>
<td>The <strong>Property Editor</strong> view presents the properties of the BIRT report elements.</td>
</tr>
</tbody>
</table>

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

There are many editors in DataQuant 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 DataQuant 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.

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

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

If you close a perspective, it will stay closed until you re-open that perspective. Each time a DataQuant 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:
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.

To open a view:
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 DataQuant sessions.

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 DataQuant 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.

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 context 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.

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:
1. Switch to the perspective that you want to configure.
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 DataQuant for Workstation, you can restore a perspective to its original layout.

To restore a perspective to its original layout:
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 for users
The Preferences dialog is used to set user preferences that will apply to specific functions in the current DataQuant application interface session.

The Preferences window 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.

To specify user preferences for the current DataQuant application interface session:
1. Select View > Preferences. The Preferences window opens.
2. Select General to specify the language, orientation direction, locale settings, and object default actions.
3. Select Appearance to specify the default fonts that will be used for query results headings and columns; the default font for classic reports; and the strings that will be used for displaying and entering null and default values.
4. Select Dashboard Runtime to set the Display null values as setting to determine how null values are displayed in the dashboards.
5. Select Data Management to specify the default connection information that will be used to produce BIRT reports. (DataQuant for Workstation only) See the BIRT Charting Programmer Reference and BIRT Programmer Reference for more information.
6. Select Global Variables to change the default values for system global variables and define user global variables.
7. Select **Help** to specify default values for how help information is displayed or change the interface and port specification for the internal server that the help system uses. (DataQuant for Workstation only)

8. Select **JDBC Libraries** to manage the JDBC files and drivers that are available to the application interface.

9. Select **LOBs** to specify options that will control retrieving and saving LOB data.

10. Select **Log** to activate a log file that will track information useful in debugging.

11. Select **Report Design** to set report design specifications for BIRT reports. (DataQuant for Workstation only)

    See the *BIRT Report Developer Guide* for more information.

12. Select **Server-Side File System** to set root export directory for objects.

13. Select **Themes** to customize the look and feel of the WebSphere window. (DataQuant for WebSphere only)

14. Select **Third-Party Libraries** to specify any third party libraries.

15. Select **Visual Designer** to specify preferences that will effect the creation of visual projects. (DataQuant for Workstation only)

16. Click **Restore Defaults** to restore preferences to their default state.

17. Click **Apply** to implement any changes that you have made to the settings in the preferences pages.

18. Click **OK** to close the Preferences window box.

### Setting general preferences

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

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

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 DataQuant 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** drop-down list the language that will be used to display numbers, currencies, times and dates in the current DataQuant 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 DataQuant 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. 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.

7. Select **Always run in background** to hide the Progress dialog from view for long running operations.

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 the default fonts that will be used for query results headings and columns; the default font for classic reports; and the strings that will be used for displaying and entering null and default values.

To specify the appearance user preferences for the current DataQuant application interface session:

1. Select **View > Preferences**. The Preferences window opens. Select **Appearance** from the tree.

2. Select from the **Headings of query results** drop-down list the font that will be used for all query results headings.

3. Select from the **Columns of query results** drop-down list the font that will be used for all query results column data.

4. Select from the **Classic reports** drop-down list the font that will be used for all classic reports.

5. Specify a string in the **Display null values as** field that DataQuant will use to display null values that are retrieved from a database data source.

6. 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.

7. 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.

8. 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 **DataQuant** 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**.

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.

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

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 Map API key field specify the Google Map API key that you will use to enable usage of the Google Map layout object in visual dashboards.
• 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.

**Setting data management preferences**

If you are using DataQuant for Workstation, you use the Data Management page of the Preferences dialog to specify the default connection information that will be used to produce BIRT reports.

To specify the connectivity user preferences for the current DataQuant application interface session:

1. Select View > Preferences. The Preferences dialog opens. Select Data Management from the tree.
2. Click Apply to save your preferences choices.
3. Click Restore Defaults to restore the default preference values.
4. Click OK to close the Preferences dialog.

**Setting help preferences**

If you are using DataQuant 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.

To change the web browser adapter setting:

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.

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:
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.
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.

DataQuant 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 DataQuant. DataQuant 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:

1. Select **View > Preferences** to open the Preferences window. Select **JDBC Libraries**. The JDBC Libraries page opens. All of the JDBC drivers and their associated JAR files that you have already set up are listed. They are organized by library.

2. To add new JDBC driver information for a type of database that is not listed, 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. If you are using DataQuant for Workstation, add the location information for the JDBC driver files to the library. Click **Add JARS**. The Add JARS 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.

7. Click **Apply**. The JDBC library information that you have specified is saved.

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**. 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 DataQuant 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**. The Edit JAR URL window opens. Specify the name and location in the **Edit JAR location** field.
   - To remove a JDBC driver library, select an existing JDBC driver library from the list and click **Remove**. The library, including all the JAR files that it contains, is removed.
   - If you are using DataQuant for Workstation, to remove a JAR file from a JDBC driver library, select an existing file from the list and click **Remove**. 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.
Setting LOB preferences

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

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:

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 DataQuant 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 DataQuant processing information.

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 DataQuant session:

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 directory location preference for objects generated by Export actions

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

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

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

To set the export directory preference:
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 or by event actions such as Export to PDF and Export to Excel, they will be exported to the 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 report design preferences
If you are using DataQuant for Workstation, you use the Report Design page of the Preferences dialog to set report design specifications for BIRT reports.

To specify the BIRT report design preferences:
2. Click Apply to save your preferences choices.
3. Click Restore Defaults to restore the default preference values.
4. Click OK to close the Preferences dialog.

Setting themes preferences
You use the Themes page of the Preferences dialog to change the look and feel of the DataQuant for WebSphere user interface.

If you are using DataQuant for WebSphere, visual designers can customize the user interface of the WebSphere window by authoring and uploading CSS files using the Themes page of the Preferences dialog. To customize the user interface of the WebSphere window:
1. Create or obtain the CSS file that will be used to control the styles of the different user interface objects.
2. Select View > Preferences. The Preferences dialog opens. Select Themes from the tree.
3. If a custom theme has already been defined, you can select it from the Themes drop down menu.

4. Specify a name for the theme in the Theme Name field.

5. Specify the CSS file that will define the theme by clicking the Browse button that is next to the CSS File field.

6. Navigate to the CSS file that you want to use and click Open. Control returns to the Add Theme dialog.

7. Click OK. The Add Theme dialog closes and the new theme is listed in the Theme text box.

8. To remove a theme, select the theme that you want to delete and click the Remove button.

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.

**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 DataQuant.

To specify the third party library preferences:


2. The third-party software whose location has been defined to DataQuant is listed by name, package name and location. In most instances, the location information for third party libraries is specified when DataQuant 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 on 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 DataQuant for Workstation, you use the Visual Designer page of the Preferences window to specify preferences that will effect the creation of visual projects.

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

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 DataQuant 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.
Discovering the contents of a workspace

The Workspaces view displays the contents of a workspace.

To discover the contents of a workspace:

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 context 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 DataQuant 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 context menu. The menu choices depend on what you have selected.

**New**   Opens a context 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:

- 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 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 dashboard that you have selected in the Workspaces or Repository Explorer views to render in the following modes: Rich Client, HTML, Optimized HTML, Flash, or PDF.

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 parent 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.
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.

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

To manage repository folders:
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 context 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 context 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 context menu, select Rename. The folder is renamed.

Filtering data source objects

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

To filter data source objects:
1. From the Workspaces view, right-click a data source that you want to filter. From the context 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.

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:

1. Right-click the object in the Workspaces view. Select Copy from the context 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 context 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 context menu, select Delete. The link is deleted.
4. To rename a link, right-click the link in the Workspaces view. From the context 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.

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:

1. Right-click anywhere in the Workspaces view to open the context 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.

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:

1. Right-click an item in the repository, select Properties from the context 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. (DataQuant 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. (DataQuant 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.

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

To move workspace objects:

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 context 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.

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:
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 context 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 context 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 context 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 DataQuant 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.

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

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.

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 context 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.

---

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 **Query Editor**.

To open a database table:

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.
   - **Query Editor** to display the table in the **Diagram Query** editor. When the table is opened with the **Query Editor**, a query is created and displayed in the **Diagram Query** editor. The query is not run and results are not returned. Users can modify the query using the **Diagram Query** 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.

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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve All</td>
<td>Select this option to retrieve all rows from the table.</td>
</tr>
<tr>
<td>Find</td>
<td>Select this option to search the table for specific data. Enter the search criteria in the Find dialog and click Find next.</td>
</tr>
<tr>
<td>Insert Row</td>
<td>Select this option to insert a table row.</td>
</tr>
<tr>
<td>Delete Row</td>
<td>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.</td>
</tr>
<tr>
<td>Edit Row</td>
<td>Select this option to edit row data.</td>
</tr>
<tr>
<td>Commit</td>
<td>Select this option to manually commit the changes you have made to the table.</td>
</tr>
<tr>
<td>Rollback</td>
<td>Select this option to undo changes that you have made to the table.</td>
</tr>
<tr>
<td>Immediate Commit</td>
<td>Select this option to enable the immediate commit function, which results in changes to the table being committed immediately.</td>
</tr>
</tbody>
</table>

Editing content directly in the table

You can edit table data directly in the Table Editor in the following ways in:

- Placing your cursor in the cell containing the information you want to edit, right-clicking and selecting Edit from the context menu.
  
  In the Cell value window, enter your changes and click OK. Click OK again to confirm that the updates will committed to the database.
  
  Click Cancel if you want to rollback 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 context 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 context menu.
   The row and its content are displayed in an Edit Row window.
c. Make changes to data in the Value of the Edit Row window and click OK.
d. Click OK to apply your updates and commit changes to the database, or click Cancel to rollback the changes.

- Inserting a row:
a. Select Insert Row from the Results menu.
   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 Rollback 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 context 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 row conditions for query results.
• Default Editor to display the table using the editor that was last used.

**Specifying sort conditions for tables**

You can specify the sort conditions that determine the order that the rows are displayed in tables.

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.
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.

Specifying row conditions for tables

If you are using DataQuant 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 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.

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. The following operators are available from the drop-down list:

- Equal To
- Less than
- Less than or equal to
- Greater than
- Greater than or equal to
- Between
- Starting with
- Ending with
- Containing
- NULL

7. Specify the condition for which to check the rows using the Right side fields. Only the rows that meet the specified conditions will be displayed in the table. You can use the expression builder button to help you build the conditions. The Expression Builder supports the SQL syntax of the associated database. You can click Import Conditions From File to specify that the row conditions will be supplied from a file. You can click Import conditions via Executing SQL Query to specify that the row conditions will be supplied from a set of query results. You can use the Remove Condition or Remove All Conditions buttons to remove obsolete 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.

### 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.

Open the database table that you want to edit on the Results tab of the Table Editor.

To edit a database table using special registers:

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 28 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.

Creating personal repositories

DataQuant 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.

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

For example, in a Windows operating system, the personal repository is saved under the following paths:

- For Windows XP or earlier:
  C:\Documents and Settings\[UserName]\Application Data\IBM\DataQuant for Workstation\Personal Repositories
- For Windows Vista or later:
  C:\Users\[UserName]\AppData\Roaming\IBM\DataQuant for Workstation\Personal Repositories

Note: The Application Data (AppData) folder is a hidden folder and will not appear in Windows Explorer unless you configure Windows Explorer to display hidden files and folders.

For other operating systems, the file is similarly located in the workstation’s personal settings directory.

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

To create a personal repository, perform the following procedure:

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 DataQuant you must be connected to a repository.

A repository is a centralized storage area that was created by your DataQuant administrator. It is the place where your objects such as queries, forms, and reports can be saved. It is also where DataQuant 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 DataQuant, 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:
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. Right-click the repository connection name for the repository to which you want to connect.
3. Select Connect. You are connected to the repository.

Administrators are responsible for creating repository connections for each user of DataQuant. If necessary, each DataQuant 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 DataQuant 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 DataQuant 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.

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

To specify a new user ID and password:
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 DataQuant 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 DataQuant 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 DataQuant 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. DataQuant will use this user information the next time it connects to the repository or data source.

---

### Changing user interface configuration information

The DataQuant 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.

Advanced DataQuant 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 DataQuant for Workstation users can change the user interface configuration information.

### Setting up JDBC driver information

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

DataQuant 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 file

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

DataQuant 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 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 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:

1. Select **File > Import > Connection and JDBC library settings**. The Import Connection and JDBC Library Settings window opens.

2. Specify where to find the JDBC library settings file in the **Settings file** field. You can use **Browse** to locate the file.

3. Select the **JDBC library settings** check box.

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**.

**Setting JDBC preferences**

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

DataQuant 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 DataQuant. DataQuant 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:

1. Select **View > Preferences** to open the Preferences window. Select **JDBC Libraries**. The JDBC Libraries page opens. All of the JDBC drivers and their associated JAR files that you have already set up are listed. They are organized by library.

2. To add new JDBC driver information for a type of database that is not listed, 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. If you are using DataQuant for Workstation, add the location information for the JDBC driver files to the library. Click **Add JARS**. The Add JARS 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.

7. Click **Apply**. The JDBC library information that you have specified is saved.
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**. 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 DataQuant 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**. The Edit JAR URL
     window opens. Specify the name and location in the **Edit JAR location** field.
   • To remove a JDBC driver library, select an existing JDBC driver library from
     the list and click **Remove**. The library, including all the JAR files that it
     contains, is removed.
   • If you are using DataQuant for Workstation, to remove a JAR file from a
     JDBC driver library, select an existing file from the list and click **Remove**.
     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.

**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 DataQuant administrator.

DataQuant 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 DataQuant.

To import the repository connections settings file:

1. Select **File > Import > Connection and JDBC library settings**. The Import
   Connection and JDBC Library Settings window opens.

2. Specify where to find the repository connections file in the **Settings file** field.
   You can use **Browse** to locate the file.

3. Select the **Repository connections** check box.

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. You can view the repository connections that are available in the **Repositories Connections** view. To connect to the repository, navigate to the **Repositories Connections** view, right-click the connection information for the repository to which you want to connect, and select **Connect**.

**Creating the repository connection for DataQuant**

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

DataQuant 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:

1. Open the Create New Repository Connection wizard by right-clicking anywhere in the **Repositories 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 DataQuant for Workstation and the server (workstation) that runs DataQuant 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. (DataQuant 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 DataQuant for Workstation.
   - **Web service repository** to create a repository connection to a shared repository through the DataQuant 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 DataQuant 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.

**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.

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:

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 Connection type radio group, select **JDBC** or **JNDI**.
   * **If you select JNDI:**
     - In the **Data Source** name field, specify the JNDI data source name and location.
     - Click **Advanced** to specify any driver-specific connection string keywords and their values that will be used for the database that is hosting the repository storage tables.
       For information on any keywords that are required by a selected database driver, consult the specific driver's documentation.
   * **If you select JDBC:**
     - In the **JDBC Driver** field specify the library for the JDBC driver that will be used to access the database that hosts 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 click the ellipsis (...) to open the JDBC Libraries window where you can create additional JDBC driver libraries.

**Note:** If you work with a DB2 for z/OS database, you can use the Direct DRDA® Connection. In contrast to JDBC libraries, the Direct DRDA Connection comes with the product and does not require additional driver installation.

**Note:** The Direct DRDA Connection is supported only for DB2 for z/OS Version 8, Version 9, and Version 10. Do not use it for DB2 LUW or DB2 for iSeries.

See the Installing and Managing PDF for information about using DRDA to connect to database repositories and data sources.
• 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 window to form the correct string. To do so, click **Build URL** to open the Build URL by URL Template window. 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.

• 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 driver, consult the specific driver’s documentation.

• If you use the **Direct DRDA Connection**, bind packages by clicking **Bind Packages**.

5. 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.

6. 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.

7. 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.

   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.

   Using this feature assumes that you selected **User ID and password are required** and that you have selected **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.
8. Click OK to close the User Information window.

9. Select **Hide from web service clients** to exclude this connection from the list of available connections in the **Repository connection name** field in DataQuant for WebSphere.
   This check box is displayed in DataQuant for WebSphere user interface only. If you are working in the DataQuant 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.

10. In the **Repository name** field, select the repository to connect to. If no repositories are listed, click **Refresh**.

11. Click **Finish**. The repository connection is listed in the **Repository Connections** view. If you selected **Connect immediately**, you are connected to the repository that you selected.

**Creating file-based repository connections:**

If you are using DataQuant 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.

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

To complete the creation of a file-based repository connection:
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 is listed in the **Repository Connections** view. If you selected the **Connect immediately**, you are connected to the repository that you selected. This repository connection information is entered in the connections settings file.

**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.

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 DataQuant for Workstation.

DataQuant for Workstation users can connect to repositories via a centralized machine running the DataQuant 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 setup and start the network repository service.

**Note:** When you create a network repository connection in DataQuant 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:

1. Verify with the DataQuant 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 is listed in the **Repository Connections** view.

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

**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 DataQuant for WebSphere web services API using a HTTP or HTTPS connection.

You can used 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 DataQuant for Workstation.

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

1. Ensure that DataQuant 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 DataQuant for WebSphere whose web services API will be used for this connection. This is the URL address that was created when the instance of DataQuant for WebSphere was deployed.
6. Click Refresh. The available repository connections that have been defined for the instance of DataQuant 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 is listed in the Repository Connections view.
   If you selected Connect immediately, you are connected to the repository that you selected. This repository connection information is entered in the connections settings file.

Currently, when the DataQuant 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 DataQuant for WebSphere client opens.

For example:

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

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.
Each DataQuant 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:

1. Open the **Repository Connections** view. Each repository connection that is available to you is listed in the **Repository Connections** view. This list includes all the repository connections that you have created as well as those that may have been supplied to you by the administrator.

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 context 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 context 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 context 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 context 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 context 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.

**Creating web links to data objects**

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

If you are using DataQuant for Workstation, the Web Link wizard is available only when there is an active web repository connection.

To create a web link to data objects:

1. Select **File > New > Other > Web Link** to open the Web Link wizard.

   **Note:** You can also access this wizard from the repository tree context menu option **Create Web Link**. If accessing the wizard using this method, the **Repository Object** field and **User Defined Parameters** are pre-filled automatically.
2. In the **Repository Object** field, select the object to which you want to create a web link (to be opened using the **Open from Repository** button).

3. Specify settings for the following **Predefined Parameters**:
   - The **Toolbar** check box indicates whether to show or hide the toolbar. By default the check box is selected, which means the toolbar is shown.
   - The **Run** check box indicates whether to run the object when it is opened. By default the check box is selected, which means the object is run when it is opened.
   - The **Environment** check box indicates whether to use the default environment to run the object. By default, the check box is not selected. This means the default environment will be used 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 check box is disabled if there are no environments.
   - The **Renderer mode** check box is available for visual reports or dashboards and it determines the mode in which visual object is rendered. The default render mode is Flash.

4. Specify **User Defined Parameters** to represent the selected repository object’s parameters (For example, variables).
   - You can add parameters if they are needed.
   - All parameters can be sorted by name.
   - Parameters that you add can be included in the link by selecting **Include in URL** check box.
     - The URL is formulated as follows:
       \[ \text{http://}\{\text{host}\}:\{\text{port}\}/\{\text{app\_context}\}/g?\text{name}=value \]

5. Click **Next** to display the second page of the Web Link wizard.
   - On this page you can complete the following actions:
     - Edit existing logins and passwords to the data sources
     - Use the **Add** button to add logins and passwords to the data sources
     - Use the **Delete** button to remove logins and passwords
   - In the **Web link name** field, enter the name of the current object which you want to use in the URL. For example, if you wanted to convert **STAFF** to **staff**, the value:
     \[ \text{http://}\{\text{host}\}:\{\text{port}\}/\{\text{app\_context}\}/g?\text{STAFF} \]
   - becomes:
     \[ \text{http://}\{\text{host}\}:\{\text{port}\}/\{\text{app\_context}\}/g?\text{staff} \]
   - If there is a link with the same name, the system prompts you to change the name. You can change the name explicitly by using the **Web link name** field.
   - Leaving the **Web link name** field empty results in a new name automatically. However, to override the name in the database you will need approval from the administrator.
   - If an administrator approves this change, the link is overwritten with the name entered in the **Web link name** field.

6. Click **Create** to generate a web link.
   - If the specified link name already exists, you must either rename it, or leave the **Web link name** field empty to create the name automatically.

7. Click **Run** to open the created web link in the browser.

8. Click **Finish** to close the Web Link wizard.
Changing your DB2 password

You can change your DB2 database password using the Change Password window.

Users are sometimes required to change their DB2 passwords on a regular basis to adhere to security measures that have been set up by their organizations. The Change Password window allows you to change your DB2 password yourself, without the aide of your database administrator.

**Note:** This feature is only valid for DB2 data sources with type four connections that use the DB2 JCC driver or Direct DRDA connectivity.

To change your DB2 password:

1. In the **Repository Explorer** view, navigate to the DB2 data source that has the password that you want to change and right-click the data source. A context menu opens.
2. Select **Properties** from the context menu. The Properties for [data source] window opens.
4. Select the **User ID and password are required** check box. You can change the following check box options:
   - **Apply logon to subsequent connections during application session** (This is the default selection)
   - **Allow users to save passwords**
   - **Allow users to change passwords**
5. Click the **Set User Information** button. The Set User Information for [data source] window opens.
6. Enter the user name in the **User:** field.
7. Enter the password in the **Password:** field.
8. Click the **Change** button. The Change Password window opens.
9. Type the old password in the **Old Password** field.

   **Note:** For security reasons, the **Old password** field will always be blank when the Change Password window opens, even if the **Remember the password** check box is selected.
10. Type the new password in the **New password** field.
11. Retype the new password in the **Confirm new password** field.
12. Select the **Remember the password** and/or **Use this information for every data source I connect to** check boxes.
13. Click **OK**. The Change Password window closes and the new password is confirmed.

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 so on) to the memory on the workstation client or on the web server if you are using DataQuant for WebSphere.

This task describes how to configure repository caching.

**Considerations:**
Caching happens in local memory on your machine if you are using DataQuant for Workstation and in memory on the web server if you are using DataQuant 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 DataQuant for Workstation, the repository cache is cleared whenever you shut down your machine or close the application.

For DataQuant for Workstation, end-users can set repository caching parameters for the version of the application running on their machine.

For DataQuant for WebSphere, repository caching is shared between all connections made with the same Repository Connection login.

For DataQuant for WebSphere, you must have administrator authorization in order to change repository connection settings.

To configure repository caching:

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 described below:

- **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 DataQuant for Workstation and on the web server if you are using DataQuant 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.

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.
Chapter 3. Creating new objects or repository items

DataQuant supplies numerous wizards to help you create new objects.

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:
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. Expand the DataQuant Objects folder to view the wizards that are available to create the querying and reporting objects.
3. Expand the Repository folder to view the wizards that are available to create objects that pertain to repositories.
   The wizards that are most frequently used to create DataQuant objects are also listed outside their folders for access convenience.
4. From the DataQuant Objects folder you select one of the following wizards:
   - Web Link to create short URLs to DataQuant objects. You can use these URLs to open the objects directly in web browsers.
   - ER Diagram to create a new Entity Relationship (ER) diagram.
   - Folder to add an ER diagram folder to the data source.
   - Form to create a new form that can be used to generate a classic report.
   - Java Script Module to create a Java Script module and open the Java Script code editor.
   - OLAP Query to create a new OLAP query.
   - Procedure to create a new procedure.
   - Prompt Hierarchy to create a prompt hierarchy.
   - Query to create a new SQL query.
   - Query using the Draw Query wizard to create a new query using the Draw Query wizard.
   - Visual Dashboard to create a new dashboard visual project. (DataQuant for Workstation only)
   - Visual Dashboard Template Category to create a new visual dashboard template category. (DataQuant for Workstation only)
   - Visual Report to create a new visual report project. (DataQuant for Workstation only)
   - Visual Report Template Category to create a new visual report template category. (DataQuant for Workstation only)
5. From the Environments folder you can choose one of the following wizards:
   - Data Source Mapping to add a data source to the environment.
   - Environment to create a new environment.
6. From the Repository folder you can choose one of the following wizards:
   - Folder to add a folder to your repository or workspace.
   - Link to create a shortcut to an object.
   - Personal Repository to create a new personal repository.
• Repository Connection to create a new repository connection.
• Workspace to create a new repository workspace.

7. If you are using DataQuant for Workstation, from the Business Intelligence and Reporting Tools folder you can choose one of the following wizards:
• Library to create a new BIRT report library object.
• Report to create a new BIRT report.
• Template to create a new BIRT report template object.

8. The following wizards are available from the Connection Profiles folder:
• Connection profile to create a new connection profile.
• Connection Profile Repository to create a new connection profile repository.

9. The following wizards are available from the Repository folder only if you have the Administrator component installed:
• Relational Data Source to add a relational data source to your repository.
• Personal Repository to create a new personal repository on behalf of a user.
• Multidimensional Data Source to add a multidimensional data source to your repository.
• Virtual Data Source to add a virtual data source to your repository.
• Shared Repository to create a new shared repository.
• Shared Repository Storage to create new shared repository storage.

10. From the Other folder, you can choose one of the following wizards:
• Scheduled Job to create a new scheduled job.

11. 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.

---

**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 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 Diagram Query editor.

### Creating queries using SQL

You can create queries by typing SQL statements in the SQL Query editor.

This task describes how to create queries by using SQL Query. It assumes that you are familiar with SQL and SQL syntax.

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 SQL Query editor provides coloring support for your SQL statements.

To create a query using the SQL Query editor:

1. Select File > New > Query. The Create New Query wizard opens. Specify the name of the query in the Name field and where the query will be saved in the Data Source field. Click Finish. Click the SQL tab.
2. Type the text of the SQL statement in the input text area. You are responsible for ensuring that the SQL statements that you type are syntactically correct.
3. You can type multiple SQL statements in a single query window. Each statement will run and multiple result sets (one for each statement) will be returned. You must include a semicolon (;) between each SQL statement.

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.

**Note:** If you are using MySQL V5.1 you can have a valid SELECT statement without tables being specified or with a pseudo table DUAL and the Add New button for Columns is always enabled in the Prompted tab.


The following MySQL attributes (keywords) in the SELECT statement are supported:

**Note:** For the MySQL V5.1 database, these keywords are supported by but cannot be viewed or edited by using the Prompted or Diagram query editors. These keywords can be viewed and edited in SQL tab only:

- HIGH_PRIORITY
- STRAIGHT_JOIN
The general syntax of the SELECT statement is:

```sql
SELECT
    [HIGH_PRIORITY]
    [STRAIGHT_JOIN]
    [SQL_SMALL_RESULT | SQL_BIG_RESULT | SQL_BUFFER_RESULT ]
    [SQL_CACHE | SQL_NO_CACHE ]
    [SQL_CALC_FOUND_ROWS]
    select clause
    [FROM
    table clause
    [WHERE clause]
    [GROUP BY clause]
    [HAVING clause]
    [ORDER BY clause]
    [LIMIT clause]
    [FOR UPDATE | LOCK IN SHARE MODE]
```

For example: `SELECT 1 + 2` is a valid SELECT statement.

Additionally, the following clauses are supported but cannot be viewed or edited in the Prompted or Diagram query editors directly; they can be viewed or edited in SQL tab only:

- GROUP BY
- HAVING
- LIMIT
- FOR UPDATE
- LOCK IN SHARE MODE

4. Click the Run Query toolbar button. The query is run against the currently connected to data source. Results are returned to the interface. The query results view is identified with a tab labeled Results. You can switch between the multiple views by clicking the corresponding tabs.

5. Once you have obtained query results you have several options.

- 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 dialog.
- Generate reports.
- Export the query results to numerous formats.

6. The SQL query object stays open until you close it. Upon closing, if you want to run the query object again you must save it. Click the Save button to open the Save object window. You can save the query object to a file, the QMF catalog, or the repository.

Creating visual queries using SQL

You can create visual query objects in DataQuant.

This task describes how to create visual queries using SQL.
To create visual queries using SQL:

1. Open the Create New Visual Query wizard by selecting File > New > Other and expanding the DataQuant Objects folder or by clicking the New Visual Query icon on the toolbar.
2. Select Visual Query from the list of DataQuant 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 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 are responsible for ensuring that the SQL statements are syntactically correct.
7. Click the Run Query icon on the toolbar.
   The visual query is run against the currently connected to 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.
8. Once you have obtained query results you have several options.
   - 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 or the repository.

**Using Content Assist for queries and visual queries**

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

To use Content Assist while typing a SQL statement:

1. Depending on whether you are creating a query or a visual query, perform one of the following steps:
   a. For queries, select File > New > Query. The Create New Query wizard opens. Specify the name of the query in the Name field and where the query will be saved in the Data Source field. Click Finish. Click the SQL tab.
   b. For visual queries, select File > New > Other and expand the DataQuant Objects folder. Select Visual Query from the list of DataQuant 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 \textit{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 \texttt{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 \texttt{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 a column, function, or table reference, and the partially typed keyword is not ambiguous, Content Assist will automatically complete the keyword. For example, when typing \texttt{SELECT * FROM Q.STAFF}, the STAFF table is automatically completed since it is the only table under the Q schema that begins with the letters ST.

6. Select an option from the drop-down list. The selected text is automatically inserted into the SQL statement.

\textbf{Using Parameter Hints for queries and visual queries}

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

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

1. Depending on whether you are creating a query or a visual query, perform one of the following steps:
   a. For queries, select \textit{File > New > Query}. The Create New Query wizard opens. Specify the name of the query in the \textit{Name} field and where the query will be saved in the \textit{Data Source} field. Click \textit{Finish}. Click the \textit{SQL} tab.
   b. For visual queries, select \textit{File > New > Other} and expand the \textit{DataQuant Objects} folder. Select \textit{Visual Query} from the list of DataQuant Objects.

2. Specify the name of the query in the \textit{Name} field and where the query will be saved in the \textit{Data Source} field. Click \textit{Finish}.

3. Click the \textit{SQL} tab.

4. Type the first part of a parameterized function, which includes the name of the function and the left parenthesis. For example, \texttt{SUM(}.

5. Press \textit{Ctrl+Shift+Spacebar}. A ToolTip opens that displays an example of how to complete the function. For example, if \texttt{SUM(} is displayed, the ToolTip will display \texttt{SUM(<numeric-expression>)}.

6. Use the displayed hint as a reference when completing the parameterized function.

\textbf{Creating queries using the Prompted Query editor}

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

To create a new SQL query using the \textit{Prompted Query} editor:

1. Select \textit{File > New > Query} to open the Create Query wizard.
   a. Specify the name of the query in the \textit{Name} field and specify where the query will be saved in the \textit{Data Source} field.
   b. Click \textit{Finish}.
   c. Click the \textit{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.

If you are running a version of DB2 that supports temporal data (for example, DB2 z/OS v10), when a table that has been configured for temporal data is added to the prompted query, the Time Period icon is enabled. If there are multiple tables in the query, the Time Period icon is enabled only when the table that is configured for temporal support is selected.

Clicking the Time Period icon launches a window in which you can specify the points in time as either system time or business time (depending on the type of time for which the table has been configured). Only one of the system time or business time pair will be enabled for any given table.

Temporal data support provides a historical perspective to table data. Historical data can often serve a specific business case. For example, a business might require access to information about the level of insurance coverage that was in place at the time of a claim several months in the past.

A table with temporal data support stores and records information about the period of time when a row is valid with respect to system time, business time, or both. System time, or transaction time, is when the transaction is recorded. Business time, or valid time, is when the data is valid with respect to information about the current real world business data.

If the DBA specified the tables and columns that require temporal data support when they are created, DB2 automatically maintains the history whenever an update is made to the data.

To specify a time period, click the Time Period icon and specify values for **System Time** and **Business Time**.

a. The drop-down list to the right of the For System Time and For Business Time labels provide SQL reference options.

b. Select As Of, From, or Between. Consult your SQL reference documentation for details on the usage of these options.

c. Set the associated date by clicking the date and time icon.

If you do not need the period you have specified, simply clear the corresponding check box. If you remove a table or edit its owner or name, then all the specified time periods will be lost.

The **As Of** option creates a temporal specification in the SQL of a FOR statement `<SYSTEM_TIME | BUSINESS_TIME> AS OF <selected timestamp>`

The **From** option creates a temporal specification in the SQL of a FOR statement `<SYSTEM_TIME | BUSINESS_TIME> FROM <selected timestamp>`

The **Between** option creates a temporal specification in the SQL of a FOR statement `<SYSTEM_TIME | BUSINESS_TIME> BETWEEN <selected timestamp1> AND <selected timestamp2>`

• **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. 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 (SQL, Prompted, Diagram, Layout, Preview, and Results) 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.

4. 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.

### 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 or the Diagram Query editor. You can also use this dialog to change the tables that have been specified in the query.

To add a table to a query when you are building the query using the Prompted Query or Diagram Query editor:

1. Open the Tables window from the Prompted Query or Diagram Query editor:
   - If you are using the Prompted Query editor, click the Add Table toolbar button in the Tables pane.
   - If you are using the Diagram Query editor, right-click anywhere in the editor and select Add Table.

   **Note:** You can also drag tables from the Workspaces or Repository Explorer views to the Tables pane.

2. Identify the table that you want to include in the query. Type the table owner in the Table owner field.

3. Type the table name in the Table name field.

4. 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.
5. 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.

6. Click **Add**. The table is added to the query. The name of the table appears in the Tables pane of the **Prompted Query** editor or a diagram of the table is inserted in the **Diagram Query** editor.

**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.

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:
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.

The step to joining tables is joining columns.

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

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:
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.

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

**Specifying query results columns**
You must specify the columns that will be included in the query results.

You use the Columns window to specify what columns will be included in the query results. By default, all the columns from a table that is included in the query are included in the query results. In the case where you have multiple tables included in the query, all the columns from each table will be included.

To specify the columns that will be included in the query results:
1. Open the Columns window by clicking the Add Column button in the Columns pane of the Prompted Query editor.
   Each table that has been 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 of the table’s 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 on columns are implemented as system column headers or column text. Column headers are used when displaying or printing query results.
   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 on 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 will be used to evaluate a column and determine whether or not it will be included in the query results.
Click the ellipsis (...) to open an expression builder. 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. From the Summary function list, select how the column will be summarized in the query results.

You can specify the following summary options:

- **(None)** - No summarization will be applied.
- **Average** - The average of all values in the column will be included in the summary field.
- **Count** - A count of all the rows will be included in the summary field.
- **Maximum** - The maximum value that was in the column will be included in the summary field.
- **Minimum** - The minimum value that was in the column will be included in the summary field.
- **Sum** - The sum of all the values in the column will be 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 will be displayed exactly as it appears in the Column field.

5. Click Add.

The column will be included in the query results and listed in the Columns pane of the Prompted Query editor.

The Add button becomes unavailable and the Columns window remains open. Repeat these steps for each column that you want to include in the query results.

Select another column, the Add button becomes available.

6. Click the Change button if you are making a change to a column that has already been added to the query results.

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

7. Click Close when you have finished 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.

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:

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.

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:
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.

   You cannot use any of the spatial predicates unless the left and right sides have spatial data types. The following operators are available from the drop-down list:
   - Equal To
   - Less than
   - Less than or equal to
   - Greater than
   - Greater than or equal to
   - Between
   - Starting with
   - Ending with
   - Containing
   - NULL

7. Specify the condition for which to check the rows using the **Right Side** fields. Only the rows that meet the specified conditions will be selected for the query results.

   You can use the expression builder button to help you build the conditions.
   - Click **Import Condition From File** to specify that the row conditions will be supplied from a file.
   - Click **Import conditions via Executing SQL Query** to specify that the row conditions will be supplied from a set of query results.
   - You can use the **Remove Condition** or **Remove All Conditions** buttons to remove obsolete 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 queries using the Diagram Query editor

If you are using DataQuant for Workstation, you can build simple to complex queries visually using the 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 SQL statements. The Diagram Query editor is not available in DataQuant for WebSphere.

To create a query using the Diagram Query editor:

1. Select File > New > Query. The Create New Query wizard opens. Specify the name of the query in the Name field and where the query will be saved in the Data Source field. Click Finish. Click the Diagram tab. The Diagram Query editor window is divided into two sections. From the top section you will specify the tables that will be in the query and the join conditions for those tables. From the bottom section you will specify the columns, sort conditions and row conditions for the query results.

2. You can add a table to the query in one of the following ways:
   - Right-click in the top section of the Diagram window to open a context menu. Select Add Table from the menu. The Tables window opens where you can select one or more tables to include in the query. A visual representation of each table that you add appears in the upper portion of the Diagram Query editor window.
   - Drag tables from the Workspaces view to the top section of the Diagram Query editor window. A visual representation of each table that you add appears in the upper portion of the Diagram Query editor window.

3. To remove a table from the query, click the Close toolbar button, and the table is removed from the editor window and the query.

4. To edit time period information in tables that support temporal data (for example, DB2 z/OS v10), click the Edit button adjacent to the Sort and Close button.
   - Hover over the table title to make these buttons visible.

5. When you include more than one table in a query you can specify how the tables will be linked. Those specifications are called the join conditions. From the Diagram Query editor window, select a column from one table and drag it to a column in another table. A connecting line is drawn from the column in the first table to the column in the second table. You can set up multiple join conditions.

6. You can edit the join conditions by right-clicking the join line that appears between any two tables. Select Change Join from the context menu to change the join conditions. The Join Tables window opens. You can select the new join conditions. If you want to delete the join condition, select Remove Join from the context menu.

7. The lower half of the Diagram Query editor window contains the query results column table where you will specify the columns that will be included in the query results and the sort and row conditions that will be applied to the query results. All columns of all the tables that have been selected for the query are automatically included in the query results column table. You will see the entry <all columns> listed in the table.

8. You can select individual columns to be included in the query results in one of the following ways:
Drag a column in one of the tables in the upper part of the Diagram Query editor window to a column in the query results column table.

Right-click a column in one of the tables in the upper part of the Diagram Query editor window. A context menu opens. Select **Add**.

The `<all columns>` entry is removed from the query results column table. The column name is added. Repeat this process for each column that you want included in the query results. The columns will appear in the query results in the same order as they appear in the query results column table. You can remove a column from the query results by right-clicking the column in either the upper part of the Diagram Query editor window or in the query results column table and selecting **Remove** from the context menu.

9. For each column that you have included in the column table, you can apply sort and row conditions using the column table fields. Double-click one of the column's cells to specify a value for one or more of the following fields:

- **Field** cell specifies the name of the current column. After double-clicking the **Field** cell, click the down arrow. All column names from all the tables are listed. You can select another column name to replace the current column.
- **Table** cell specifies the name of the table that includes the current column.
- **Display name** cell specifies the name that will be used as the column header in the query results.
- **Include** cell indicates that the column will be included in the query results.
- **Aggregation** cell specifies what type of aggregation, if any, will be applied to the column. After double-clicking in the **Aggregation** cell, click the down arrow. The types of aggregation that you can apply are listed.
- **Sort order** cell specifies whether this column will be used to sort the rows of data in the query results. After double-clicking the **Sort order** cell, click the down arrow. From the drop-down menu, select **Ascending** to sort in ascending (lowest-to-highest) order, select **Descending** to sort in descending (highest-to-lowest) order, or select **not sorted** to exclude the column from any sort condition.
- **Key sequence** cell specifies the sort order sequence for the column. The column with the lowest number will be sorted first. Select the sequence number from the drop-down list.
- **Row conditions** cell specifies a condition that must be met for a row of data to be included in the query results. The condition that you specify here applies specifically to the column value. For example, if you had a column called `Number` and you entered a row condition of `> 10`, the only rows of data included in the query results will be 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.

10. You can specify additional row conditions using the **Additional row conditions** field. You would use this field to specify row conditions that affect more than a specific column in the query results. For example, you would use this field if you wanted a condition to be met that includes a column of data that is available in a table but not included in the query results. The **Additional row conditions** field has two columns. The first column contains the operator. The second column has the condition. Click the first cell (just below the header cell). If you have not specified any row conditions for any columns in the column table, the **IF** operator is available. If you have
specified one or more conditions for any column in the column table, the
operator AND and the operator OR are available. Click the second column to
specify the condition.

11. To specify whether or not duplicate rows will be included in the query results,
select the Include duplicate rows check box.

12. When you complete building the query, click the Run Query toolbar button.
The query that has been created is run against the currently connected to data
source. Results are returned.

13. The query results are identified with a Results tab. You can switch between
the multiple views of a query (SQL, Prompted, Diagram, Layout, Preview, and
Results) 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.

14. The query object stays open until you close it. Upon closing, if you want to
run the query object again you must save it. Click the Save toolbar button to
open the Save object window. Depending on the permissions that have been
granted to your user ID, you can save the query object to a file, a workspace,
the QMF catalog, or the repository.

Creating visual queries using the Diagram Query editor
If you are using DataQuant for Workstation, you can build simple to multi-variable
queries visually using the 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
SQL statements. The Diagram Query editor is not available in DataQuant for
WebSphere.

To create a visual query using the Diagram Query editor, perform the following
procedure:
1. Select File > New > Other and expand the DataQuant Objects. Select Visual
Query and click Next.

   Note: You can also click the New Visual Query ( ) toolbar button.

   a. Specify the name of the visual query in the Name field and where the
   visual query will be saved in the Data Source field.

   b. Click Finish.

   c. Click the Diagram tab

      The Diagram Query editor window is divided into two sections. From the
top section you will specify the tables that will be in the query and the join
conditions for those tables. From the bottom section you will specify the
columns, sort conditions and row conditions for the query results.

2. Add a table to the query by right-clicking in the top section of the Diagram
window and clicking Add Table from the context menu.

The Tables window opens and you can select one or more tables to include in
the query.
A visual representation of each table that you add appears in the upper portion of the Diagram Query editor window.

3. To remove a table from the query, click the Close toolbar button, and the table is removed from the editor window and the query.

4. When you include more than one table in a query you must specify how the tables will be linked. Those specifications are called the join conditions. From the Diagram Query editor window, select a column from one table and drag it to a column in another table. A connecting line is drawn from the column in the first table to the column in the second table. You can set up multiple join conditions.

5. You can edit the join conditions by right-clicking the join line that appears between any two tables. Select Change Join from the context menu to change the join conditions. The Join Tables window opens. You can select the new join conditions. If you want to delete the join condition, select Remove Join from the context menu.

6. The lower half of the Diagram Query editor window contains the query results column table where you will specify the columns that will be included in the query results and the sort and row conditions that will be applied to the query results. All columns of all the tables that have been selected for the query are automatically included in the query results column table. You will see the entry <all columns> listed in the table.

7. To select individual columns to be included in the query results, perform the following steps:
   a. Right-click a column in one of the tables in the upper part of the Diagram Query editor window.
   b. From the context menu select Add.
      The <all columns> entry is removed from the query results column table. The column name is added. Repeat this process for each column that you want included in the query results. The columns will appear in the query results in the same order as they appear in the query results column table. You can remove a column from the query results by right-clicking the column in either the upper part of the Diagram Query editor window or in the query results column table and selecting Remove from the context menu.

8. For each column that you have included in the column table, you can apply sort and row conditions using the column table fields. Double-click one of the column's cells to specify a value for one or more of the following fields:
   - The Field cell specifies the name of the current column. After double-clicking in the Field cell, click the down arrow. All column names from all the tables are listed. You can select another column name to replace the current column.
   - The Table cell specifies the name of the table that includes the current column.
   - The Display name cell specifies the name that will be used as the column header in the query results.
   - The Include cell indicates that the column will be included in the query results.
   - The Aggregation cell specifies what type of aggregation, if any, will be applied to the column. After double-clicking in the Aggregation cell, click the down arrow. The types of aggregation that you can apply are listed.
   - The Sort order cell specifies whether this column will be used to sort the rows of data in the query results. After-double clicking in the Sort order cell, click the down arrow. From the drop-down menu, select Ascending to
sort in ascending (lowest-to-highest) order, select **Descending** to sort in
descending (highest-to-lowest) order, or select **not sorted** to exclude the
column from any sort condition.

- The **Key sequence** cell specifies the sort order sequence for the column. The
column with the lowest number will be sorted first. Select the sequence
number from the drop-down list.

- The **Row conditions** cell specifies a condition that must be met for a row of
data to be included in the query results. The condition that you specify here
applies specifically to the column value. For example, if you had a column
called **Number** and you entered a row condition of > 10, the only rows of
data included in the query results will be 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.

9. You can specify additional row conditions using the **Additional row
conditions** field. You would use this field to specify row conditions that affect
more than a specific column in the query results. For example, you would use
this field if you wanted a condition to be met that includes a column of data
that is available in a table but not included in the query results. The
**Additional row conditions** field has two columns. The first column contains
the operator. The second column has the condition. Click in the first cell (just
below the header cell). If you have not specified any row conditions for any
columns in the column table, the IF operator is available. If you have
specified one or more conditions for any column in the column table, the
operator AND and the operator OR are available. Click in the second cell to
specify the condition.

10. To specify whether or not duplicate rows will be included in the query results,
select the **Include duplicate rows** check box.

11. When you complete building the query, click the **Run Query** button. The
query that has been created is run against the currently connected to data
source. Results are returned.

12. The query results are identified with a **Results** tab. You can switch between
the multiple views of a query (SQL, Diagram, Design, Display, and Results) 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 charts.
- Generate quick reports.
- Export the query results to numerous formats.

13. The 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. Click the **Save
button to open the Save object window. Depending on the permissions that
have been granted to your user ID, you can save the visual query object to a
file, a workspace, or the repository.

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

1. Open the Tables window from the Prompted Query or Diagram Query editor:
   • If you are using the Prompted Query editor, click the Add Table toolbar button in the Tables pane.
   • If you are using the Diagram Query editor, right-click anywhere in the editor and select Add Table.

   **Note:** You can also drag tables from the Workspaces or Repository Explorer views to the Tables pane.

2. Identify the table that you want to include in the query. Type the table owner in the Table owner field.

3. Type the table name in the Table name field.

4. 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.

5. 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.

6. Click Add. The table is added to the query. The name of the table appears in the Tables pane of the Prompted Query editor or a diagram of the table is inserted in the Diagram Query editor.

**Editing join conditions for multiple tables**

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

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

1. Right-click the line that joins the table diagrams in the top section of the Diagram Query 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:
   • **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**. The new join conditions are saved.

**Creating 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.

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

To create a query using the Draw Query wizard:

1. Open the Create New Query Using the Draw Query wizard by clicking the **Draw Query** toolbar button.

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**.

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**.

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.

**Listing data source tables**

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

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.

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

<table>
<thead>
<tr>
<th>Object type</th>
<th>Available prompts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>SQL prompts</td>
</tr>
<tr>
<td>Visual query, dynamart</td>
<td>SQL prompts, simple prompts, and prompt hierarchies</td>
</tr>
<tr>
<td>Analytical query, compound dynamart</td>
<td>Shared prompts that consist of simple prompts</td>
</tr>
</tbody>
</table>

The following topics contain the detailed information about adding prompts:

**Working with SQL prompts**

You use SQL prompts to limit the data retrieved from a query or visual query.

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 query or a visual query:

1. Open the SQL tab.

   *Note:* You can specify a prompt in the Conditions column on the Build tab of the 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.

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 visual 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, when you run a query against company staff data, you may want to display data for a specific department only. You add a simple prompt to the query and each time you run it, you will be asked to enter the department name which data you are interested in.

When you create a query with a substitutional parameter and use simple prompts with preset values like department name, these values are added automatically to the query and you are not prompted to enter them manually. Query results will contain data for the preset department name only.

To work with simple prompts use either Manage prompts window or the Prompts pane on the Design page of the 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.

Use the Simple Prompt window to specify a prompt and add it to a query.

To add a simple prompt to a query:

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, the table on the Prompts 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 Add 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. Click **OK** to close the Simple Prompt window.

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.

**Saving prompt values to a query:**

The Prompt values that you enter when running a query can be saved to the query.

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 input 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:

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 input the
prompts value anytime you run the query. If the check box is selected, you will be able to 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 the query to a repository or file system.

**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 DataQuant 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.

To create a new prompt hierarchy:
2. Expand the DataQuant Objects folder, select Prompt Hierarchy, and click Next.
3. Specify a name for the prompt hierarchy in the Name field, and click Finish. The Prompt Hierarchy editor opens.

Note: The Prompt Hierarchy editor window consists of the following sections:
- **Data Source**
  This section includes user interface controls for specifying the location from which the prompt hierarchy will draw its data.
- **Hierarchy levels**
  This section provides user interface controls for constructing prompt hierarchy levels and specifying their properties.
- **Prompt hierarchy preview**
  This section provides the area from which you can preview the prompt hierarchy.

4. From the Data Source section of the Prompt Hierarchy editor, specify the source of the data from which the Prompt Hierarchy will be created. The data source can be either embedded or linked. The linked option is selected by default.
   a. For linked data source:
      - Click the ellipsis (...) at the end of the Path field.
The Select a Wizard wizard opens.

- Select **Open from Repository**, and click **Next**.
- Drill down through the relational data sources to select the desired repository object, and click **Finish**.

You are returned to Prompt Hierarchy editor. The Path field is populated in the top section of the Prompt Hierarchy editor window.

5. From the Hierarchy levels section of the Prompt Hierarchy editor window, use the Add Level icon to add levels to the hierarchy. As you add and select each level, the following hierarchy level properties become modifiable:

   - **Level name** - The name of the hierarchy level.
   - **Display value** - The column that is displayed when the user is prompted to enter the variable value. For example, there are two columns that contain associated department values. One contains the department names, and the other contains the internally recognized department numbers. Setting the department name column as the display value provides a more user friendly way to enter variable values.
   - **Return value** - The column that returns the variable value that the user provides to the query. 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.
   - **Order by** - The column by which the query results are ordered. For example, if the “NAME” column is set as the order by column, the query results will be sorted alphabetically.

Use the Move level up icon to move selected levels up in the hierarchy.
Use the Move level down icon to move selected levels down in the hierarchy.

6. From the Prompt hierarchy preview section of the Prompt Hierarchy editor window, use the Refresh icon to preview the prompt hierarchy.

7. The Prompt Hierarchy editor window stays open until you close it. Upon closing, the system prompts you to save changes. Click **Yes**. The Save Prompt Hierarchy wizard opens. You can save the prompt hierarchy to either a repository or a file.

8. To save the prompt hierarchy to a repository, select **Save to Repository** and click **Next**. The Save to Repository wizard opens.

9. Navigate to the repository location where you want to save the prompt hierarchy, specify a name in the **Name** field, optionally specify a comment in the comment field and click **Finish**. The Save to Repository wizard closes and the prompt hierarchy is saved.

**Editing existing prompt hierarchies:**

You can modify an existing prompt hierarchy by using the Prompt Hierarchy editor window.

To edit an existing prompt hierarchy:

1. From the Workspaces or Repository Explorer view, navigate to the prompt hierarchy that you want to edit and double-click it. The prompt hierarchy opens in the Prompt Hierarchy editor.
2. From the Source of data section of the Prompt Hierarchy editor window, specify the source of the data. The data source can be either embedded or linked. The linked option is selected by default.
3. For linked data source, click the ellipsis (...) at the end of the Path field. The Open wizard opens.

4. Select Open from Repository, and click Next. The Open from Repository wizard opens.

5. Drill down through the relational data sources to select the repository object, and click Finish. The Open wizard closes and the Path field is populated in the top section of the Prompt Hierarchy editor window.

6. From the Hierarchy levels section of the Prompt Hierarchy editor window, use the Add Level icon to add hierarchy levels. As you add and select each level, the following hierarchy level properties become modifiable:
   - Level name where you can specify the level name.
   - Display value where the drop-down list is populated with all of the available display values.
   - Return value where the drop-down list is populated with all of the available return values.
   - Order by where the drop-down list is populated with all of the available order values.

   Use the Move level up icon to move selected levels up in the hierarchy.

   Use the Move level down icon to move selected levels down in the hierarchy.

7. From the Prompt hierarchy preview section of the Prompt Hierarchy editor window, use the Refresh icon to preview the prompt hierarchy.

8. The Prompt Hierarchy editor window stays open until you close it. Upon closing, the system prompts you to save changes. Click Yes. The Prompt Hierarchy editor window closes and the changes to the prompt hierarchy are saved.

**Working with shared prompts**

Use shared prompts to apply filtering to analytical queries and compound dynamarts.

In analytical queries, you use prompt groups that include shared child query prompts or newly added simple prompts.

To manage shared prompts in analytical queries or compound dynamarts:

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 move a prompt from the default group to a different group of shared prompts:
   a. From the Available prompts list, select a prompt that you want to add to a group.
   b. From the Prompt Group list, select a group to which you want to add the prompt.
   c. Click Add to Group. The selected prompt is added to the prompt group.

   **Note:** Only prompts of one type can be added to the same prompt group.

4. To remove a prompt from a prompt group:
a. From the Prompt Group list, select the prompt that you want to remove.
b. Click Detach Prompt. The selected prompt is detached from the prompt group.

5. 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.

6. 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.

7. If you work in the Manage Prompts window, click OK to close the window.

Specifying substitution variables

Substitution variables are used to input 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, 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:

```sql
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:

1. Open a query.
2. Type this SQL statement: `SELECT * FROM Q.STAFF WHERE DEPT = &MIN_DEPT`
3. Run the query. The Enter Substitution Variable Values 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.

Try experimenting with substitution variables by replacing values in the SELECT and FROM clauses. See what results your queries return.
Setting fonts for SQL statements

If you are using DataQuant for Workstation, you can set unique font attributes for the SQL statement text.

To set unique font attributes:
1. Display SQL statements in the workstation query editor.
2. Open 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.

Query menu

The Query menu becomes available when the active object in the editor is a query or visual query.

The following menu commands are available:

Run
   This command runs the active query.

Prepare
   This command validates the active query by checking the syntax and existence of objects in the query. Internal calculations are performed but no results are returned.

Cancel
   This command cancels the active query that is currently running.

Reformat Text
   This command reformats the SQL text of a query. It will neaten the appearance of the text, capitalize SQL keywords, and shorten queries to less than 79 characters per line, a requirement for saving a query in a QMF catalog. (DataQuant for Workstation only)

   The line width value specified in the Reformat Query Options section of the General Preferences page is applied to the SQL query text when you run the Reformat Text command.

Comment
   Add comments to a query so that part of the query is not run. Comments are signified by "--" in the query window. (DataQuant for Workstation only)

Uncomment
   Deletes the comment marks "--" so that the text is run with the query. (DataQuant for Workstation only)

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 window opens.
Set User Information
Opens the Set User Information window where you specify the user ID and password that will be used to connect to the data source where you want the active query to run.

Set Row Limit
Opens the Set Row Limit window, where you can specify the number of rows that the active query will fetch from the database.

Set Font
Opens the Font window where you can change the display font attributes for the SQL statements of the active query.

Confirm Stored Procedure Parameters
Defines the behavior of Confirm Stored Procedure Parameters window in which you input and confirm the data types, modes, and values of parameters for stored procedures called in a visual query. If this option is selected, you are prompted to input and confirm the stored procedure parameters anytime you run a visual query with a stored procedure called in it. If this option is cleared, the Confirm Stored Procedure Parameters window opens only if at least one of the parameters is not specified. By default this option is selected.

Convert Query to Visual Query
This menu item is available only for queries and it transforms a query into a visual query.

Transfer to
This menu item is available only for queries and by selecting the necessary options, you can transform your query into Drill-down path, Forecast or Visual Report.

Add New Statement
This menu item is available when Prompted or Diagram tabs are selected. It adds a new result set to the current query.

Remove Current Statement
This menu item is available when Prompted or Diagram tabs are selected. It deletes the current result set from the query.

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.

Using DataQuant for Workstation and DataQuant for WebSphere you can create an OLAP query for multidimensional data sources that is queried using MDX and accessed using XMLA.

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.

All the data cubes that are available for your use are listed in your workspace.

To retrieve cube data for an OLAP query:
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.
   - **Preview** to preview the OLAP query results.

**Opening existing OLAP queries**

You can open existing OLAP queries that are included in your workspace.

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.

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:
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.

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.

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:
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.

**Filtering cube data**

Filtering enables you to limit the amount of data that is retrieved from a cube.

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:

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.

**Working with OLAP query results**

You can format the OLAP query results directly in the editor window.

The options that are available to you are accessible from the **Results** menu or a context menu opened by right-clicking the editor window. The options available depend on what you have selected in the editor window.

With OLAP query results in the available editor window you can:

• Use the **Font** option to change the font for a measure or dimension. The Font page of the Layout Properties window opens.
• Use the **Format** option to change the format for a measure or dimension. The Format page of the Layout Properties window opens.
• Select the **Show Summaries** option to view summary information for a measure or dimension.
• Select the **Show Total Summaries** option to view only totals for a measure or dimension summarized information.
• Use the **Drill Down** option to view detailed data for a summarized dimension.
• Use the **Drill Up** option to view only the summary data for a dimension and its corresponding measures.
• Use the **Reset Formatting** to restore the default formatting to the OLAP query results.
• Use **Autofit** to expand and contract the column size to fit the data.

**OLAP Query menu**

The **OLAP Query** menu becomes available when the active object in the editor is an OLAP query.

The following menu commands are available:

**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 window opens.
Set User Information
Opens the Set User Information window where you specify the user ID and password that will be used to connect to the data source where you want the active query to run.

Creating analytical queries
Analytical queries allow you to combine data from multiple queries from the same or differing data sources into one result set.

Using analytical queries, you can combine query results from multiple queries that span different data sources into one query result set. There are several principle query types that you can incorporate into an analytical query structure tree:

- **Append Query** - Append queries combine query results from two queries into one query result set. You can specify how the columns will be paired up across the two result sets.
- **Join Query** - Join queries join the result sets from two queries into one unified result set by joining one or more columns together, much like a join operation in SQL.
- **Crosstab Query** - Crosstab queries provide 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 Query** - Normalized queries "normalize" the text returned in a query result set. Normalizing query text facilitates query processing, such that executions of the same query with different parameters can be easily compared with one another and aggregated together.
- **Conditional Grouping Query** - With Conditional Grouping queries you can create aggregated summaries of result set data based on conditional expressions.
- **Column Filter Query** - With Column filter queries you can add a calculated column before or after existing columns and enter filtering functions.

Multiple queries and tables can be added into the analytical query structure tree to generate a comprehensive result set that encompass data from many different sources. The different options of each query type allow you to customize how this data is displayed.

To create an analytical query:
1. Select **File > New > Other**. The New window opens.
2. Expand the **DataQuant 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 **Structure** query editor with a query pre-loaded in the **Analytical query structure** pane.
5. To add a query or table object to the analytical query structure, select the query node under which you want to add the new query, click **Add**, and then select the type of the object that you want to add. The new query or table object is displayed under the selected query node.

**Note**: You can also add a table saved in a data file to an analytical query, by selecting **Add Table** from the query item context menu and opening the required file.
6. To change a query to another query type, select a table or query, click **Change** to and select the new query type.

7. To rename a table or query, select it and click **Rename**. The Rename Query window opens.

8. Specify the new name for the query or table in the **Query name** field and click **OK**. The Rename Query window closes and the query or table is renamed.

9. To detach a table or query, select it and click **Detach**. A copy of the query or table opens in a separate editor. Any changes that are made have no effect on the analytical query that it was taken from.

   **Note:** You cannot detach the table that was added from a data file.

10. To open and edit a query or table in a separate editor, select it and click **Open in Separate Editor**. The query or table opens in a separate editor. Any changes that are made have a direct effect on the analytical query that it was taken from.

   **Note:** You cannot open the table that was added from a data file in a separate editor.

11. To add a query that is currently open in a separate editor, right-click an empty slot in the analytical query structure tree, select **Add From Opened**, and select one of the available queries.

### Adding an append query

You use append queries to add selected result set columns from different queries to the final result set.

Append queries allow you to combine result set columns from different queries 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 add an append query to an analytical query:

1. Create an analytical query and display it in the **Structure** query editor.
2. Right-click in the **Analytical query structure** pane and select **Add Append Query** from the context menu.
3. Right-click the first node named `<none>` and select **Add Query** from the context menu. The Open window opens.
4. Navigate to and select the query that you want to set as the first inner query and click **Finish**. The Open window closes and the query’s columns are displayed in the **First** list of the **Item editor** pane.
5. Right-click the second node named `<none>` and select **Add Query** from the context menu. The Open window opens.
6. Navigate to and select the query that you want to set as the second inner query and click **Finish**. The Open window closes and the query’s columns are displayed in the **Second** list of the **Item editor** pane.
7. 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 process for each append condition that you want to add.

8. If you are populating the append query with queries that contain similar names, you can attempt to automatically create the appropriate append conditions by clicking Automatch. If there are any possible automatches, they are displayed in the Result columns list.

Note: Clicking Automatch will erase any append conditions that you previously set.

Adding a join query
You use join queries to join result set columns from different queries and tables.

Join queries allow you to join columns from different queries 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.

To add a join query to the analytical query:
1. Create an analytical query and display it in the Structure query editor.
2. Right-click in the Analytical query structure pane and select Add Join Query from the context menu.
3. Right-click the first node named <none> and select Add Query from the context menu. The Open window opens.
4. Navigate to and select the query that you want to draw left columns from and click Finish. The Open window closes and the query's columns are displayed in the Left Columns list of the Item editor pane.
5. Right-click the second node named <none> and select Add Query from the context menu. The Open window opens.
6. Navigate to and select the query that you want to draw right columns from and click Finish. The Open window closes and the query's columns are displayed in the Right Columns list of the Item editor pane.
7. Specify the query result columns that you want to join together.
   a. Select the type of join that you want to add from the Join Type drop-down list.
   b. Select a column from the Left Columns list.
   c. Select a column from the Right Columns list.
   d. Click Add Join Key above the Join Columns list. The new join condition is added to the Join Columns list.
   e. Repeat this process for each join condition that you want to add.
8. To add a column to the result set from the left or right columns, select a column and click Add to Result Columns.
   To move the added result column up or down, select the column and click the To Up or To Down button above the Result Columns list.
9. To add all of the available columns, click Add All Columns above the Result Columns list.

Adding a crosstab query
You use crosstab queries to create aggregated summaries of result set data across grouped columns.

Crosstab queries are useful for isolating and displaying the aggregated values of grouped result set columns. For example, you want to know the sum of the
salaries of the employees in each department of a company, grouped into three job types. With a crosstab query, you can create a result set that displays a row for each department and a column for each of the three job types. Each cell of the result set displays the sum of the salaries of the employees in given department who perform the given job type. For example, you can quickly find the sum of the salaries of all of the clerks in department number ten.

To add a crosstab query:
1. Create an analytical query and display it in the Structure query editor.
2. Right-click in the Analytical query structure pane and select Add Crosstab Query from the context menu.
3. Right-click the first node named <none> and select Add Query from the context menu. The Open window opens.
4. Navigate to and select the query that you want to draw crosstab results from and click Finish. The Open window closes and the query is displayed in the Analytical query structure pane.
5. Click Add in the Group pane. The Select Columns window opens.
6. Select the column that you want to display in the side group and click OK. The Select Columns window closes and the column is displayed in the Columns table.
7. Click the value in the Aggregation column and select how you want to aggregate the column from the drop-down list.
8. Select the column that you want to display in the top group from the Grouping column drop-down list.
9. Select the column that will provide the descriptions for the top group columns from the Description column drop-down list.
10. Click Add in the Crosstab pane. The Select Columns window opens.
11. Select the column that will provide the values in the cells of the crosstab result set and click OK. The Select Columns window closes and the column is displayed in the Value columns table.
12. Click the value in the Aggregation column and select how you want to aggregate the values in the cells of the result set from the drop-down list.

**Adding an analytical query**

You can add an analytical query to the Analytical query structure tree at any point.

Previously created analytical queries can be added to the analytical query structure tree.
1. Create an analytical query and display it in the Structure query editor.
2. Right-click in the Analytical query structure pane and select Add Analytical Query from the context menu.
3. Navigate to and select the analytical query that you want to add and click Finish. The Open window closes and the analytical query is displayed in the Analytical query structure pane.

**Adding a normalize query**

Normalization of query result data allows you to collapse multiple columns into two columns within a new result set.
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.

**Table 6. Original result set data**

<table>
<thead>
<tr>
<th>Region</th>
<th>Revenue</th>
<th>Units Sold</th>
<th>Net Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>250</td>
<td>452</td>
<td>87</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>290</td>
<td>538</td>
<td>92</td>
</tr>
<tr>
<td>Europe</td>
<td>320</td>
<td>675</td>
<td>120</td>
</tr>
</tbody>
</table>

Normalization allows us to transform the data into the following two forms:

**Table 7. Normalized query results**

<table>
<thead>
<tr>
<th>Region</th>
<th>KPI</th>
<th>KPI Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>Revenue</td>
<td>250</td>
</tr>
<tr>
<td>North America</td>
<td>Units Sold</td>
<td>452</td>
</tr>
<tr>
<td>North America</td>
<td>Net Income</td>
<td>87</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>Revenue</td>
<td>290</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>Units Sold</td>
<td>538</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>Net Income</td>
<td>92</td>
</tr>
<tr>
<td>Europe</td>
<td>Revenue</td>
<td>320</td>
</tr>
<tr>
<td>Europe</td>
<td>Units Sold</td>
<td>675</td>
</tr>
<tr>
<td>Europe</td>
<td>Net Income</td>
<td>120</td>
</tr>
</tbody>
</table>

**Table 8. Alternate normalized query results**

<table>
<thead>
<tr>
<th>Region</th>
<th>KPI</th>
<th>KPI Value</th>
<th>Units Sold</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>Revenue</td>
<td>250</td>
<td>452</td>
</tr>
<tr>
<td>North America</td>
<td>Net Income</td>
<td>87</td>
<td>452</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>Revenue</td>
<td>290</td>
<td>538</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>Net Income</td>
<td>92</td>
<td>538</td>
</tr>
<tr>
<td>Europe</td>
<td>Revenue</td>
<td>320</td>
<td>675</td>
</tr>
<tr>
<td>Europe</td>
<td>Net Income</td>
<td>120</td>
<td>675</td>
</tr>
</tbody>
</table>

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 in this sample) and the column that indicates the original column (KPI Value) 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:

**Table 9. Crosstabbed query results for furniture sales**

<table>
<thead>
<tr>
<th>Order Date</th>
<th>Gross Profit (CHAIRS)</th>
<th>Gross Profit (SOFAS)</th>
<th>Gross Profit (MISC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>19 November 2008</td>
<td>160</td>
<td>452</td>
<td>87</td>
</tr>
<tr>
<td>23 November 2008</td>
<td>0</td>
<td>680</td>
<td>181</td>
</tr>
<tr>
<td>26 November 2008</td>
<td>120</td>
<td>642</td>
<td>0</td>
</tr>
</tbody>
</table>

Normalizing the query reverses the cross-tabulation to produce query results that will look like the original query results before the cross-tabulation was applied. The Normalized query results appear like the example shown below:

**Table 10. Normalized query results for furniture sales**

<table>
<thead>
<tr>
<th>Order Date</th>
<th>Category</th>
<th>Gross Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>19 November 2008</td>
<td>CHAIRS</td>
<td>160</td>
</tr>
<tr>
<td>19 November 2008</td>
<td>MISC</td>
<td>87</td>
</tr>
<tr>
<td>19 November 2008</td>
<td>SOFAS</td>
<td>452</td>
</tr>
<tr>
<td>23 November 2008</td>
<td>MISC</td>
<td>181</td>
</tr>
<tr>
<td>23 November 2008</td>
<td>SOFAS</td>
<td>680</td>
</tr>
<tr>
<td>26 November 2008</td>
<td>CHAIRS</td>
<td>120</td>
</tr>
<tr>
<td>26 November 2008</td>
<td>SOFAS</td>
<td>642</td>
</tr>
</tbody>
</table>

To add a normalize query:

1. Create an analytical query and display it in the Structure query editor.
2. Right-click in the Analytical query structure pane and select Add Normalize Query from the context menu.
3. Right-click the first node named <none> and select Add Query from the context menu. The Open window opens.
4. Navigate to and select the crosstabbed or aggregated query that you want to normalize and click Finish. The Open window closes.
5. In the Dimensions section, click Add above the No Transform list box, select the columns that you want to include without alterations and click OK. These columns will be displayed without transformation and will contain repeated values for each row of normalized columns. In the example above, the Order Date column is the No Transform column.
6. Click Add above the Columns list box. The Dimension Column window opens.
7. Specify the name and type of the column in the Name field and Type drop-down list. This is the new column that will display the names of the normalized columns. In the example above, this is the Category column.
8. Specify the names of the normalized columns that will be displayed in the Constants table and click OK. In the example above, these 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.
9. Click Add above the Values list box. The Value Column window opens.
10. Specify the name and type of the column in the **Name** field and **Type** drop-down list and click **OK**. This is the new column that will display the values of the normalized columns. In the example above, this is the **Gross Profit** column.

11. In the **Normalize Columns** section, click **Add**, select the columns that you want to normalize and click **OK**. These are the columns that you want to move down into the specified dimension columns. In the example above, these are the **Gross Profit (CHAIRS)**, **Gross Profit (SOFAS)** and **Gross Profit (MISC)** columns.

12. Select the value of the normalized column from the **Value** drop-down list.

13. Select the category of the normalized column from the **Category** drop-down list.

14. Select the **Suppress Zeros** or **Suppress Nulls** check boxes to exclude rows from the normalized result set that contain values of zero or no values at all.

15. Once you have finished adding normalize columns, click the **Run Query** toolbar button. The query is run and the normalized results are displayed in the **Results** editor.

**Adding a conditional grouping**

Conditional groupings allow you to create aggregated summaries of result set data based on conditional expressions.

A conditional grouping lets you 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.

**Table 11. Original data**

<table>
<thead>
<tr>
<th>Department</th>
<th>Name</th>
<th>Years</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>SALES</td>
<td>Smith</td>
<td>12</td>
<td>60</td>
</tr>
<tr>
<td>SUPPORT</td>
<td>Jones</td>
<td>6</td>
<td>40</td>
</tr>
<tr>
<td>R&amp;D</td>
<td>Adams</td>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td>R&amp;D</td>
<td>Baker</td>
<td>7</td>
<td>50</td>
</tr>
</tbody>
</table>

Using a conditional grouping, the result set can be transformed into the following:

**Table 12. Conditionally grouped data**

<table>
<thead>
<tr>
<th>Department</th>
<th>Total Salary</th>
<th>Matched Employees</th>
<th>Total Employees</th>
</tr>
</thead>
<tbody>
<tr>
<td>SALES</td>
<td>100</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>SUPPORT</td>
<td>40</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>R&amp;D</td>
<td>50</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

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 add a conditional grouping:

1. Create an analytical query and display it in the Structure query editor.

2. Right-click in the Analytical query structure pane and select Add Conditional Grouping from the context menu.

3. Right-click the first node named <none> and select Add Query from the context menu. The Open window opens.

4. Navigate to and select the query that you want to draw columns from and click Finish. The Open window closes.

5. In the Group section, click Add. The Select Columns window opens.

6. Hold down the Ctrl key, select the column names that you want to group aggregated results by, and click OK. The Select Columns window closes and the group columns are displayed in the Columns table. Aggregated values will be broken down for each row of a given group column.

7. Specify how a group column will be aggregated by clicking in the Aggregation column and selecting an aggregation method from the drop-down list.


9. 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.

10. Select a filter building method and click Next. The next page of the wizard depends on your previous selection.

11. 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.

12. In the Columns table, select the column that you want to filter from the Source column drop-down list. This column will provide the values that will be filtered and aggregated across the group columns.

13. Select the filter that you will use with this column from the Filter drop-down list. This list is populated with all of the filters that you built in the Filters table.

14. Select how the values will be aggregated from the Aggregation drop-down list.

15. Specify a name for the column of aggregated values in the Column name field.

16. 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 query 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 headers 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.

For more information about building filters, see:
- “Building a compound condition filter” on page 104
- “Building a free style condition filter” on page 106

### 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, DataQuant 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 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.

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

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.

**Saving objects to a file**

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

To save an object to a file:

1. If you are using DataQuant 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 DataQuant 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 desired location on your local machine.

**Saving objects in the QMF catalog**

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

To save an object to a QMF catalog:

1. To open the Save to QMF 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. 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 queries, forms, and procedures.

6. You can specify a comment in the Comment field.

7. 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.

8. Click Finish. The object is saved in the QMF catalog.
Opening saved objects

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

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

To open a saved object:
1. Select File > Open From.
2. From the context 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 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.

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

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.

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

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.

You can open QMF objects in one of two ways:
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. 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.

f. Click OK. The object opens in the editor window.
Chapter 4. Managing query results

With query results in the available editor window you can perform several formatting tasks.

Formatting the query results

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 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.

To apply formatting options:
1. By default the Properties view opens with the User perspective. 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 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.

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:
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 context 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.

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:

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.
To specify the font formatting options:

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.

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.

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:
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.

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:
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 (such as MIN, MAX, AVG, SQRT, CONCAT, SUBSTR), 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 use the Dynamart Filter Wizard to create conditional expressions that control the contents of query results.

Filters allow you to restrict the returned rows and values of a query result set to your specifications. Any number of filters can be set for each result set, with any number of expressions that give you complete control over the content of the grid. To filter query results:

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 two ways; either use a table of options to assist you in building your filter, or use the expression designer to build a free style filter.
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.

If you want to use multiple conditions to filter your query results, you can build more than one filter from a table of options in the Dynamart Filter Wizard. To build a compound condition filter:
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 are like the specified value are included.
   - **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.
   - **From Hierarchy** - If a prompt hierarchy is associated with the query, values are included based on the hierarchy selections.

4. Select the type of value that you want to specify from the **Type1** drop-down 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 is 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.

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 DataQuant expression designer syntax. To build a free style condition filter:

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 use to filter the results in the **Expression** field.
3. Use the function categories on the right to help you construct your filter expressions.
4. Once you have finished building you filter expressions, click **Finish**. The Dynamart Filter Wizard closes and the query results are filter.

**Adding calculated columns to the query results**

Calculated columns are columns of data that you can add to the query results. You use the Calculated Columns window to add a calculated column to the query results.

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.

The type of functions supported for calculated columns vary depending on whether your results are from a query or a visual query.

- If you are adding calculated columns to query results, see *Functions for calculated columns supported in queries* in the product help for the list of supported functions.
- If you are adding calculated columns to visual query results, see *Functions for calculated columns supported in visual queries* in the product help for the list of supported functions.

The content of the added column is calculated using an expression that you define. The expression can use data from other columns, constants and variables. 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 results each time the query is run.

To add a calculated column to the query results:

1. Open the Calculated Column window in one of the following ways:
   - If you are using DataQuant for Workstation, on the Results page of the editor, right-click a column's header cell in the query results and select **Add Calculated Column Before** from the context menu.
   - With query results in the active editor, select **Results > Add Calculated Column**.
• On the Layout page of the editor, right-click a column name in the Layout Structure tree and select Add Calculated Column Before from the context menu.

2. Specify the name of the calculated column as it should appear in the query results in the Name field.

3. Type the expression that will be used to calculate the contents of the column in the Expression field.

   The expression that you build can use data from other columns, constants, variables and functions.

   You must adhere to a specific set of syntax rules when building the expression for the calculated column. 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 (such as MIN, MAX, AVG, SQRT, CONCAT, SUBSTR), and a set of logical and relational operators (such as , , =, !=). The expression used to calculate the value of the added column is saved with the query.

4. Select the type of the calculated column from the Type drop-down menu. The type determines the format of the calculated column data. Valid choices are:
   • Auto: The application automatically chooses the type of the calculated column from the given expression and the columns used.
   • Integer: Data is expressed as whole numbers.
   • Float: Data is expressed in a more expansive numeric format.
   • Decimal: Numeric data is expressed in decimal form.
   • String: Data can be expressed as complex strings.
   • Date: Data is expressed in a date format.
   • Time: Data is expressed in a time format.
   • Time Stamp: Data is expressed as both a date and a time.

5. Click OK to close the Calculated Column window. The content of the column is calculated based on the expression and added to the query results. If you selected Add calculated column from the context menu, the new column is added as the last column in the query results. If you selected Add calculated column before from the context menu, the new column is inserted in the query results before the selected column.

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.

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:

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 DataQuant 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 context menu.
From the Layout page of the editor, right-click a column name in the Layout Structure tree and select Grouping and Aggregation from the context menu.

2. From the Layout page of the available editor, right-click a column label field from the Layout structure tree. A context menu opens for the selected column on the right.

3. Select the grouping and aggregation option that you want to apply to the column.

4. 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 context 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.

You use the Export Query Results wizard to export query results.

To export query results:

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.

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.

This task describes how to send query results in an email.

1. Open the Export Query Results wizard. With a set of query results in active editor, select **File > Send To > Mail Recipient**...

   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)**

   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:

   - 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

5. 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.

   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.
6. Click **Next** to add another attachment
7. Click **Click here to add new file** label
8. Click the ellipse (...) 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:

```html
<img src="cid:[image name]"/>
```

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:

```html
<img src="cid:chartx.gif"/>
```

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.

To export query results data to a database:

1. Open the Export Query Results wizard. Select **Database**. The fields that are required to export query results to a database are displayed.
2. 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.
3. 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.
4. Optional: If you want to attach a comment to the data, type the comment in the **Comment** field.
5. 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.
6. Click **Next**. The Setup save options page opens.
7. 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 query result set rows will be retrieved and saved to the database.
   - **All - Result set only** to specify that all of the query result set rows that have been retrieved so far will be saved to the database. For example, if a query is run that returns 10,000 rows, but just 200 rows are retrieved at the time of export, only those 200 rows are saved to the database.
- **Selection** to specify that only the data, as it is currently formatted, that you have selected in the current query results will be exported. This can include calculated columns if they are selected. The query result data that you want to export must be selected before you open the Export Query Results window.

8. Specify the method by which the data is saved by selecting one of the following:
   - 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).

   Click this radio button to perform a fast save data process where the query is rerun at the database and the results of the query are inserted directly in the specified table.

   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. This method retains any ORDER BY clauses.

   When the fast save method is selected, the **Remove ORDER BY clause(s) from original SQL before re-running query** check box becomes available. Select the check box to invoke the fast save mode that strips the ORDER BY clause(s) from the query.

9. 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.

10. You can choose to export any calculated columns that have been added to the query results by checking the **Save calculated columns** check box.

11. 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.

12. 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.

To export a query result set to Microsoft Excel correctly, start each Excel formula with “=” in the query.

For example,

```sql
SELECT 45.5 AS PRODPRICE, 30 AS AMOUNT, '=A2*B2' AS PRICE
FROM SYSBM.SYSDUMMY1
```

To save query results to a file:
1. Open the Export Query Results wizard. Select **File system**. The fields that are required to export query results to a file system are displayed.
2. From the **Source** list, select the result set that you want to export. If the query you want to export contains several result sets, you can export all of its result
sets in one Microsoft Excel XLS file by selecting the All result sets option. Each of the query's result sets will be saved in a separate Excel worksheet.

**Note:** Visual queries do not support several result sets.

3. If you are using DataQuant 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 spreadsheet (*.xls)

5. If you selected any option different from XLS, 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. Click Finish. If you are using DataQuant for Workstation, a save window specific to your operating system opens. If you are using DataQuant 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.

**Table 13. Text options**

<table>
<thead>
<tr>
<th>Field</th>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What Data</td>
<td>• .txt</td>
<td>You use the following radio buttons to specify what data in the query results will be saved:</td>
</tr>
<tr>
<td></td>
<td>• .html</td>
<td>• All to specify that all of the data in the current query results will be saved including any calculated columns that have been added.</td>
</tr>
<tr>
<td></td>
<td>• .csv</td>
<td>• All - Result set only to specify that all of the data in the current query results will be saved excluding any calculated columns that have been added.</td>
</tr>
<tr>
<td></td>
<td>• .ixf</td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• .wqml</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• .xml</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• .dbf</td>
<td></td>
</tr>
<tr>
<td>Include Column</td>
<td>• .txt</td>
<td>Select this check box to include the query results column headings as the first row in the file.</td>
</tr>
<tr>
<td>Headings</td>
<td>• .html</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• .csv</td>
<td></td>
</tr>
</tbody>
</table>
Table 13. Text options (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date/Time Format</td>
<td>.txt</td>
<td>Select one of the radio buttons to specify how the date and time should be formatted in the file:</td>
</tr>
<tr>
<td></td>
<td>.html</td>
<td>- 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</td>
</tr>
<tr>
<td></td>
<td>.csv</td>
<td>the time will be formatted as (HH:mm:SS).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the time will be formatted as (HH:mm PM/AM).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the time will be formatted as (HH:mm:SS).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the time will be formatted as (HH.mm.SS).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Custom to specify a custom format for date and time values. &quot;Date:&quot; and &quot;Time:&quot; text fields accept Java format strings. (See <a href="http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html">http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html</a> for more details.)</td>
</tr>
<tr>
<td>CCSID</td>
<td>.txt</td>
<td>Use this field to select the code page to use when saving this file.</td>
</tr>
<tr>
<td>Display null values</td>
<td>.txt</td>
<td>Use this field to specify the string that will be used to display null values.</td>
</tr>
<tr>
<td>Separate columns</td>
<td>.txt</td>
<td>Use this field to specify the string that will be used to separate the columns.</td>
</tr>
<tr>
<td>Delimit values</td>
<td>.txt</td>
<td>Use this field to specify the string that will be used to delimit values.</td>
</tr>
<tr>
<td>When should values be delimited</td>
<td>.txt</td>
<td>Use this field to specify when values should be delimited. Your choices are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Never</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Only when necessary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Always</td>
</tr>
<tr>
<td>Which type of values should be</td>
<td>.txt</td>
<td>Select one or more of the following check boxes to specify what types of values should be delimited in the file:</td>
</tr>
<tr>
<td>delimited</td>
<td></td>
<td>- Column headings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Character values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Numeric values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Date/time values</td>
</tr>
<tr>
<td>Field</td>
<td>File Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Format</td>
<td>.ixf</td>
<td>Select one of the following radio buttons to specify how to format the .ixf output file. Your choices are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PC/IXF (Version 1) to export data in PC/IXF Version 1 format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PC/IXF (Version 2) to export data in PC/IXF Version 2 format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• System/370 IXF to export data in System/370 IXF format.</td>
</tr>
<tr>
<td>Add Document</td>
<td>.pdf</td>
<td>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:</td>
</tr>
<tr>
<td>Information</td>
<td></td>
<td>• Title specifies the title of the PDF document.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Author specifies the author of the PDF document.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Subject specifies the subject of the PDF document.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Keywords specifies the keywords for the PDF document. Keywords are used as search criteria.</td>
</tr>
<tr>
<td>Use only ANSI</td>
<td>.pdf</td>
<td>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.</td>
</tr>
<tr>
<td>Fonts</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 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.

CSV and TXT files are imported into the active editor where they can be saved as result sets to the repository or QMF Catalog. To import a CSV or TXT file:

1. Select File > Import. The Import wizard opens.
2. Select CSV File and click Next. The Import result set from file (*.csv, *.txt) page of the Import wizard opens.
3. Click the 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 drop-down 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 the result set’s column options in the Field definitions table and click Finish. 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 DataQuant will convert the data of the CSV or TXT file into the columns of the result set.
The separator options of the CSV File Import wizard allows you to specify how DataQuant 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:

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 quote or a double quote is 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.

By specifying the options in the Field definitions table, you can customize the structure of the imported result set:

1. Select what columns to include in the result set by checking the check boxes to the left of the Field name column.
2. Edit the name of a column by clicking a cell in the Field name column and entering the new name.
3. Specify the type of a column by clicking a cell in the Type column and specifying a new column type from the drop down menu. 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 can vary depending on the database you use.

4. Specify the length of a column's character set by clicking a cell in the **Length** column and specifying a new length from the list.

**Note:** You can track how each selection changes the layout of the result set in the Preview table.

5. 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.

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.

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.

To import data contained in IXF files into a database table:

1. Select **File > Import**. The Import wizard opens.
2. Select **Table** and click **Next**. The Import table from file (*.ixf, *.xml, *.csv, *.txt, *.dbf) 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. Specify the table owner name in the **Table owner** field.
6. Click **Next**. The Setup save options page of the Import Table wizard opens.
7. Specify **All** for the **What data?** option. Since you are retrieving data from a file rather than a query, all rows and columns will be imported.
8. Specify **Regular** for the data source **Method** option. Regular uses an insert statement for each row.
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 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.

To import data contained in a file into a database table:

1. Select **File > Import**.
2. From the list of available wizards select **Table** and click **Next**.
The Import table 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:
   - *.txt
   - *.csv
   - *.ixf
   - *.xml
   - *.dbf

4. Click **Next**.

   If you import the table contents from *.txt or *.csv files, the CSV File Import page opens.
   a. From the **Encoding** drop-down list, specify the type of encoding to use when importing the file.
   b. Click **Next**
      
      The Select separator options page opens.
   c. Specify the separator options of the CSV or TXT file.

5. Click **Next**.

   The Select results and destination information page opens. Specify the table owner name in the **Table owner** field.

6. Click **Next**

   The Setup save options page of the Import Table wizard opens.

7. Specify **All** for the **What data?** option.

   **Note:** Since you are retrieving data from a file rather than a query, all rows and columns are imported.

8. Specify **Regular** for the **data source Method** option.

   Regular uses an insert statement for each row.

9. 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.

10. Click **Finish**.

    The Import Table wizard closes and the data contained in the file is saved to the database table.

---

**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.

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 DataQuant administrator.
To retrieve or save LOB data:

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 context 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 wish 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 DataQuant administrator has granted your user ID permission, you can override the LOB resource limits that have been set for your resource limits group.

To override the LOB retrieval resource limit:
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 DataQuant for Workstation, you can use the Zoom option to view in hexadecimal or binary the contents of a cell in the query results.

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.

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 context 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 context menu. The Zoom Edit window opens.
   b. The CCSID field specifies the code page 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. The Cell value field is editable.
   d. Make the necessary changes and select OK to apply them.
   e. 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.

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. (DataQuant 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:** Visual queries do not support several result sets.

**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.

---

### Sorting query results

Use the Sort dialog to sort the query results by up to three columns of data.

To specify the columns by which the query results will be sorted:

1. With query results in the available window, open the Sort window by selecting **Results > Sort**.

2. Specify the first column that will be used to sort the query results using the **First Sort Rule** fields. You must select the column from the **Column** list, specify the type of sort in the **Type of sort** list, and select **Ascending** or **Descending** to specify the sort order.

3. Optionally, you can specify a second and a third column that will be used to sort the query results. You use the **Second Sort Rule** and **Third Sort Rule** fields to specify the column, type of sort and the sort order for each of the additional columns.

4. By default, the **Sort string based on locale rules** check box is clear 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.

5. Click **OK**. The query results are sorted and the Sort window closes.
If the DSQQW_QUERY_PRESERVE_SORT global variable is set to 1, the sorting order specified is saved to the query and will used automatically the next time you run the query. 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 dialog.

**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 (DataQuant for Workstation or DataQuant 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 DataQuant interface and the database. The DataQuant 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.

In some instances on DB2 for z/OS systems, keeping the ORDER BY clause can cause an SQL error. This error results from the DataQuant interface prepending an insert statement to the query. The original query is now a subordinate query and thus is not permitted to have an ORDER BY clause.

For example if a user enters the following query:

```
Select [Columns] from [SourceTable]
Order By [Column]
```

The query runs without error. The user then selects to save it into a given table (e.g. DestTable) using the fast save mode. The DataQuant interface prepends a line to the query as follows:

```
Insert into [DestTable]
Select [Columns] from [SourceTable]
Order By [Column]
```

This new SQL is invalid in certain DB2 for z/OS systems because the original query is now a subordinate query and thus is not permitted to have an ORDER BY clause.

**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 Setup 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 safe 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 DataQuant interface remove all ORDER BY clauses from the query before the query is sent to the database. Using the same example as above, 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_DA. The default value for DSQQW_FST_SV_DATA is the value zero (0).

1. Select **View > Preferences**. Expand the DataQuant 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 DataQuant 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 DataQuant interface.
Chapter 5. Generating 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.
- Graphical reports authored in the Business Intelligence and Report Tool (BIRT) report designer. (DataQuant for Workstation only)

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.

To generate a report in any of the formats:
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.
     - Create a BIRT report to create a BIRT report. (DataQuant for Workstation only)

   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.

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. DataQuant 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 DataQuant 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, DataQuant 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:

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.

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:
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. 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.
7. From this list, select the object that you want to use as the classic report data source object.
8. Click OK. The Object List window closes. Control returns to the Data Source Object window.

Exporting classic and visual reports to a file
You can export a classic or visual report to a file in text, HTML or PDF format. If your report is a classic report, you can export it as a text, HTML or PDF file. If your report is a visual report you can export it as an HTML or PDF file.

To export a classic report:
1. With a report in the available window, select File > Export. The Export window opens.
2. Select Report and click Next. The Select the report and the file destination page of the Export window opens.

3. Select the report that you want to export from the Reports list box.

4. If you are using DataQuant 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.

5. 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 visual report as an HTML file (*.htm;*.html) or a PDF file (*.pdf).

6. If you are exporting a PDF file, click the Next button. The Setup 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.

7. Click Finish.
   - If you are using DataQuant for Workstation, the report is exported to a file in the format that you selected.
   - If you are using DataQuant 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 desired 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.

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.


### 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.

To create a form:

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 fill in 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.

Some of the elements that you define in the Main section of the report can also be defined in other sections.

### Table 14. Main section elements for a classic report

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total width of report columns</td>
<td>This property displays the total width of the tabular data in the report.</td>
</tr>
<tr>
<td>Column properties</td>
<td>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 <a href="#">Column properties</a> for details.</td>
</tr>
<tr>
<td>Page</td>
<td>Use the Heading text and Footing text fields to specify the heading and footing of the page.</td>
</tr>
<tr>
<td>Final</td>
<td>Use the Text field to specify the final line text.</td>
</tr>
<tr>
<td>Breaks</td>
<td>Use the following elements to specify breaks properties:</td>
</tr>
<tr>
<td></td>
<td>• New page for break</td>
</tr>
<tr>
<td></td>
<td>Check this check box to start a new page in the report when the current break level occurs.</td>
</tr>
<tr>
<td></td>
<td>• Footing text</td>
</tr>
<tr>
<td></td>
<td>Use this field to specify the break footing text.</td>
</tr>
</tbody>
</table>
Table 14. Main section elements for a classic report (continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outlining for break columns</td>
<td>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.</td>
</tr>
<tr>
<td>Default break text (*)</td>
<td>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.</td>
</tr>
</tbody>
</table>

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:  
  
  \text{FIRST} \\  \text{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.

*Table 15. Breaks section elements for a classic report*

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New page for break</td>
<td>Check this check box to start a new page in the report when the current break level occurs.</td>
</tr>
<tr>
<td>Repeat detail heading</td>
<td>Check this check box to repeat the current detail heading with the current break level.</td>
</tr>
<tr>
<td>Blank lines before heading</td>
<td>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.</td>
</tr>
<tr>
<td>Blank lines after heading</td>
<td>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.</td>
</tr>
</tbody>
</table>
Table 15. Breaks section elements for a classic report  (continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break heading text</td>
<td>You use the buttons above the Break heading text table to add, edit, or remove lines, and also change the lines order. You use the following fields to specify the break heading text:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Line</strong>: 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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Alignment</strong>: 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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Text</strong>: Use this field to specify the text that appears in the break heading. You can use form variables and global variables in this field.</td>
</tr>
<tr>
<td>New page for footing</td>
<td>Check this check box to start a new page in the report when the current break level ends.</td>
</tr>
<tr>
<td>Put break summary at line</td>
<td>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.</td>
</tr>
<tr>
<td>Blank lines before footing</td>
<td>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.</td>
</tr>
<tr>
<td>Blank lines after footing</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Table 15. Breaks section elements for a classic report (continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break footing text</td>
<td>You use the buttons above the <strong>Break footing text</strong> table to add, edit, or remove lines, and also change the lines order.</td>
</tr>
<tr>
<td></td>
<td>Use the following fields to specify the break footing text:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Line</strong>: 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 <strong>Alignment</strong> field. Lines with the same value in the <strong>Line</strong> field overlay each other if they are longer than the report width, or if their values in the <strong>Alignment</strong> field conflict.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Alignment</strong>: Use this field to specify where horizontally on the line this text is placed. Valid alignment values are <strong>LEFT</strong>, <strong>RIGHT</strong>, <strong>CENTER</strong>, <strong>APPEND</strong>, or the number of a specific position in the line. If you specify <strong>APPEND</strong>, the text is placed immediately following the previous line of the text.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Text</strong>: 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.</td>
</tr>
</tbody>
</table>

### 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.

### Table 16. HTML variables

<table>
<thead>
<tr>
<th>HTML Variable</th>
<th>Use this variable to…</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGEn</td>
<td>Places the contents of column ‘n’ inside an HTML image reference tag, making graphic files visible in reports.</td>
</tr>
<tr>
<td>LINKn</td>
<td>Places the contents of column ‘n’ inside an HTML link tag, which is used to add links to other HTML pages.</td>
</tr>
<tr>
<td>MAILTO:n</td>
<td>Places the contents of column ‘n’ inside an HTML mail to tag for an email link.</td>
</tr>
<tr>
<td>ANCHORn</td>
<td>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.</td>
</tr>
<tr>
<td>REFn</td>
<td>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 ANCHORn variable in the report.</td>
</tr>
</tbody>
</table>
Specifying Calculations formatting options
You use the Calculations options to define report calculation expressions.

You can specify the Calculations formatting options in the Calculations table. You use the buttons above the table to add, edit or remove Calculations, and also change the Calculations order.

Use the elements described in the table below to set Calculation properties.

Table 17. Calculations option elements for a classic report

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Specify a number, between 1 and 999, that identifies the calculation expression.</td>
</tr>
<tr>
<td>Expression</td>
<td>Type a valid expression, up to 50 characters. Form variables cannot be used in this field.</td>
</tr>
<tr>
<td>Pass nulls?</td>
<td>Specify whether to pass the expression for evaluation when a variable in the definition has a null value.</td>
</tr>
<tr>
<td>Width</td>
<td>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.</td>
</tr>
<tr>
<td>Edit</td>
<td>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.</td>
</tr>
</tbody>
</table>

Specifying Columns formatting options
You use the Columns formatting options to control the appearance and formatting of columns in the report.

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 18. Columns option elements for a classic report

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Num</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Heading</strong></td>
<td>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</td>
</tr>
<tr>
<td><strong>Usage</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Indent</strong></td>
<td>Specifies the number of blank spaces to indent from the left of the column. The range is between 0 and 999.</td>
</tr>
<tr>
<td><strong>Width</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Edit</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Seq</strong></td>
<td>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 <strong>Num</strong> field. <strong>Note:</strong> The sequence number is ignored in reports using the ACROSS usage code or if you check the Automatic reordering of report columns check box.</td>
</tr>
</tbody>
</table>

**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.
You can specify the Conditions formatting options in the **Conditions** table. You use the buttons above the table to add, edit, or remove condition lines, and change the condition lines order.

Use the elements described in the table below to set condition properties.

*Table 19. Conditions option elements for a classic report*

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Specify a number, between 1 and 999, that identifies the conditional expression.</td>
</tr>
<tr>
<td>Expression</td>
<td>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.</td>
</tr>
<tr>
<td>Pass nulls</td>
<td>Specify whether to pass the expression for evaluation when a variable in the definition has a null value.</td>
</tr>
</tbody>
</table>

**Specifying Details formatting options**

You use the **Details** formatting fields to define report detail headings and body text.

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.

*Table 20. Detail section elements for a classic report*

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable</td>
<td>Specify when to enable the detail variation:</td>
</tr>
<tr>
<td></td>
<td>• <strong>YES</strong> indicates that the variation is always selected for formatting.</td>
</tr>
<tr>
<td></td>
<td>• <strong>NO</strong> indicates that the variation is never selected for formatting.</td>
</tr>
<tr>
<td></td>
<td>You can conditionally enable the variation by referring to a Form Conditions expression using the Cnn and Enn specifications.</td>
</tr>
<tr>
<td>Include column headings with detail headings</td>
<td>Check this check box to display column headings with detail headings.</td>
</tr>
</tbody>
</table>
Table 20. Detail section elements for a classic report (continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail heading text</td>
<td>You use the buttons above the <strong>Detail heading text</strong> table to add, edit, or remove lines, and also change the lines order. Specify the lines of detail heading text:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Line</strong>: 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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Alignment</strong>: 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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Text</strong>: type the text that appears in the detail heading. You can use form variables, global variables, and HTML variables in this field.</td>
</tr>
<tr>
<td>New page for detail block</td>
<td>Check this check box to start a new page in the report for the detail block.</td>
</tr>
<tr>
<td>Blank lines after block</td>
<td>Specify the number of blank lines after the last detail block line. The range is between 0 and 999.</td>
</tr>
<tr>
<td>Repeat detail heading</td>
<td>Check this check box to repeat the detail heading before each occurrence of the detail block.</td>
</tr>
<tr>
<td>Put tabular data at line</td>
<td>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.</td>
</tr>
<tr>
<td>Keep block on page</td>
<td>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.</td>
</tr>
<tr>
<td>Detail block text</td>
<td>You use the buttons above the <strong>Detail block text</strong> table to add, edit, or remove lines, and also change the lines order. Specify the lines of detail block text:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Line</strong>: 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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Alignment</strong>: 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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Text</strong>: type the text that appears in the detail block. You can use form variables, global variables and HTML variables in this field.</td>
</tr>
</tbody>
</table>
Specifying Options formatting options
You use the Options formatting options to define miscellaneous options that control the appearance of the report.

Table 21. Options section elements for a classic report

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail spacing</td>
<td>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.</td>
</tr>
<tr>
<td>Line wrapping width</td>
<td>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.</td>
</tr>
<tr>
<td>Report text line width</td>
<td>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.</td>
</tr>
<tr>
<td>Number of fixed columns in report</td>
<td>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.</td>
</tr>
<tr>
<td>Outlining for break columns</td>
<td>Select this check box to display the value in columns with the BREAK usage code when the value changes.</td>
</tr>
<tr>
<td>Default break text (*)</td>
<td>Select this check box to generate break footing text in breaks for which you did not indicate break footing text.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Function name in column heading when grouping</td>
<td>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.</td>
</tr>
<tr>
<td>Column wrapped lines kept on a page</td>
<td>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.</td>
</tr>
<tr>
<td>Across summary column</td>
<td>Select this check box to display the automatically generated ACROSS summary column, which produces additional columns that summarize (total) across the specified columns.</td>
</tr>
</tbody>
</table>
Table 21. Options section elements for a classic report (continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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.

Table 22. Final section elements for a classic report

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New page for final text</td>
<td>Check this check box to start a new report page for the final text.</td>
</tr>
<tr>
<td>Put final summary at line</td>
<td>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.</td>
</tr>
<tr>
<td>Blank lines before text</td>
<td>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.</td>
</tr>
</tbody>
</table>
Table 22. Final section elements for a classic report (continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>You use the buttons above the Final text table to add, edit, or remove lines, and also change the lines order. Specify the lines of final text:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Line</strong>: 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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Alignment</strong>: specify where the text is placed horizontally on the line. Valid alignment values are <strong>LEFT</strong>, <strong>RIGHT</strong>, <strong>CENTER</strong>, <strong>APPEND</strong>, or the number of a specific position in the line. If you specify <strong>APPEND</strong>, the text is placed immediately following the previous line of text.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Text</strong>: type the text that appears at the bottom of the report. You can use form variables, global variables and HTML variables in this field.</td>
</tr>
</tbody>
</table>

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.

Table 23. Page section elements for a classic report

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank lines before heading</td>
<td>Specify the number of blank lines before the first page heading line. The range is between 0 and 999.</td>
</tr>
<tr>
<td>Blank lines after heading</td>
<td>Specify the number of blank lines after the last page heading line. The range is between 0 and 999.</td>
</tr>
<tr>
<td>Page heading text</td>
<td>You use the buttons above the Page heading text table to add, edit, or remove lines, and also change the lines order. Type each line of page heading text.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Line</strong>: 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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Alignment</strong>: specify where to place the text horizontally on the line. Valid alignment values are <strong>LEFT</strong>, <strong>RIGHT</strong>, <strong>CENTER</strong>, <strong>APPEND</strong>, or the number of a specific position in the line. If you specify <strong>APPEND</strong>, the text is placed immediately following the previous line of text.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Text</strong>: 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.</td>
</tr>
<tr>
<td>Blank lines before footing</td>
<td>Specify the number of blank lines before the first page footing line. The range is between 0 and 999.</td>
</tr>
<tr>
<td>Blank lines after footing</td>
<td>Specify the number of blank lines after the last page footing line. The range is between 0 and 999.</td>
</tr>
</tbody>
</table>
Table 23. Page section elements for a classic report (continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page footing text</td>
<td>You use the buttons above the Page footing text table to add, edit, or remove lines, and also change the lines order. Specify each line of text printed in the page footing at the bottom of the page:</td>
</tr>
<tr>
<td>Line:</td>
<td>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.</td>
</tr>
<tr>
<td>Alignment:</td>
<td>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.</td>
</tr>
<tr>
<td>Text:</td>
<td>type the page footing text that appears at the bottom of the report.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Usage Code</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
</table>
| ACROSS     | Enables you to produce a report with horizontal control breaks. | In an ACROSS report:  
• 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. |
<p>| 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. |</p>
<table>
<thead>
<tr>
<th>Usage Code</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BREAKnX</td>
<td>Same as BREAKn, except the control column is omitted from the report.</td>
<td>Same as BREAKn.</td>
</tr>
<tr>
<td>CALCid</td>
<td>The evaluation of the calculation expression in the Form calculations component where ID equals &quot;id&quot;.</td>
<td>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.</td>
</tr>
<tr>
<td>COUNT</td>
<td>Count of the non-null values in the column.</td>
<td>This calculated value appears as a total in the report. The calculated value is formatted with the edit code K.</td>
</tr>
</tbody>
</table>
| CPCT      | Cumulative percentage each value of the column is of the current total. | • 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. | • 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. |
<p>| 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 &amp;n, where n represents the column number). |</p>
<table>
<thead>
<tr>
<th>Usage Code</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
</table>
| PCT        | Percentage each value of the column is of the current total. | • 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. | • 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. | • 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.

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.

To save a form with an associated data source object:
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.

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 BIRT reports**

DataQuant for Workstation includes the Business Intelligence Reporting Tools (BIRT), Eclipse-based reporting system.

The BIRT report format is particularly useful when there’s a need to share reporting templates or content between DataQuant for Workstation and third-party applications. By incorporating the BIRT infrastructure into DataQuant for Workstation, BIRT report authors benefit from the following robust query development, security, and content management features native to DataQuant:

- Object storage in the DataQuant repository
- Access to DataQuant queries and data sources
- Subject to the same object tracking, governing and security rule set applied to native DataQuant for Workstation content
- Capable of running in both DataQuant for Workstation and DataQuant for WebSphere.

Content developers that are already familiar with BIRT will notice the following two additions to the BIRT designer embedded within DataQuant for Workstation:
The ability to add DataQuant data sources to the report's list of data sources.

The ability to add DataQuant queries (embedded or linked) to the report's list of data sets.

For more information on working with the BIRT report designer refer to the included BIRT-specific help system.

The following steps outline the tasks that you will perform when creating or editing BIRT reports:

1. Activate a BIRT report in the workstation editor window. You can activate a BIRT report in the workstation editor window in one of the following ways:
   - Create a new BIRT report using the Display Report wizard.
   - Create a new BIRT report using the New Report wizard.
   - Open an existing BIRT report that has been stored in a repository.

2. Open the Report Design perspective. The Report Design perspective presents an editor window, several views and the menus that you will use when you work with BIRT reports. You can open the Report Design perspective by using either the Open perspective toolbar button or by selecting Window > Open Perspective > Report Design.

3. View the key components of the BIRT report as listed in the Outline view.

4. Add a DataQuant data source to a report's list of available data sources. From the Outline view, right-click on the Data Sources folder. Select New Data Source from the menu. The New Data Source dialog opens. Select the Create from a data source type in the following list radio button. Select DataQuant Data Source from the list of data source types.

5. Click Next. The New DataQuant Data Source Profile dialog opens. Select the DataQuant data source from the list of data sources. Click Finish.

6. Specify a query that will be run against the DataQuant data source to supply data for the BIRT report. Queries for a BIRT report are stored in the Data Sets folder. From the Outline view, right-click on the Data Sets folder. Select New Data Set from the menu. The New Data Set dialog opens.

7. Specify a name for the query in the Data Set Name field. This can be any name. It will identify the query in the Data Sets folder.

8. Specify the name of the DataQuant data source against which this query will run in the Data Source field.

9. Choose whether you will embed or link the query. In the Data Set Type field, select one of the following:
   - DataQuant Embedded Data Set to embed the query. An embedded query is stored within the BIRT report.
   - DataQuant Linked Data Set to set up a link to the query that is stored in a repository.

10. Click Next. Depending on your selection one of the following windows opens:
    - If you selected DataQuant Embedded Data Set, a BIRT query editor opens where you can create your query.
    - If you selected DataQuant Linked Data Set, a second page opens requesting the path to the query as it appears in the repository. Enter the path including query name.

11. Click Finish. The data set for the query has been created. To run the query and obtain query results, select the data set and drag it to the report in the editor window. The report will be populated. Click the Preview tab to view the report.
12. You can make changes to the BIRT report using the BIRT report designer. For information on editing functions see the BIRT-specific help systems.

13. Each time that you click the Preview tab you are actually running the BIRT report object and you can view how your results will appear.

14. When you have completed designing or editing the BIRT report, you can save the report in the repository.

15. BIRT provides several report viewers that allow you to view your report in different formats. You can access the viewers by selecting File > View Report from the menu. The following viewers are available:
   - View Report in Web Viewer
   - View Report as XLS
   - View Report as POSTSCRIPT
   - View Report as HTML
   - View Report as DOC
   - View Report as PPT
   - View Report as PDF

Form menu

In DataQuant, 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 DataQuant 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 6. 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.

DataQuant 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 above, then **Neural Networks** 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 above 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 DataQuant 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.

To create a forecast, perform the following 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 DataQuant 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.

### Specifying forecast data source options

You use the **Data Source** pane of the **Group** page to specify the data source of a forecast.

To create a forecast and specify data source options:

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.

### Specifying forecast date parameters

You use the **Date Parameters** pane of the **Group** page to specify the date range options of the forecast.

To specify date parameters of the forecast:

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 in the **Date column** drop-down list.

**Note:** Only date column drop-down list.

**Note:** The following formats are available for date:
- YYYYxMMxDD
• MMxDDxYYYY
• DDxMMxYYYY
• YYxMxDD
• MMxDDxYY
• DDxMMxYY

For time:
• HHxMMxSS
• HHxMM

For timestamp:
• YYYYxMMxDD HHxMMxSS
• DDxMMxYYYY HHxMMxSS
• YYYYxMMxDD HHxMM
• DDxMMxYYYY HHxMM
• YYxMxDD HHxMMxSS
• DDxMMxYY HHxMMxSS
• YYxMxDD HHxMM
• DDxMMxYY HHxMM

3. Specify the start and end dates of the forecast in the From and To calendar fields.
4. Select the time period over which to aggregate data values in the Period drop-down list.
5. Specify the number of periods beyond the To date to calculate forecast values in the Number of future periods field.

Specifying forecast grouping hierarchy options

You use the Grouping Hierarchy pane of the Group page to specify the columns that will group the forecast data.

To specify grouping hierarchy options:
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.

Specifying forecast strategy options

Use the Strategy pane of the Forecast editor to specify how the forecasted data is formatted.

To specify strategy options:
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.

---

**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.

To specify forecasting models options:
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**.

---

Chapter 6. Working with Forecasts 155
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.

**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.

To specify simple moving average parameters:
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.

To specify weighted moving average parameters:
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.

To specify single exponential smoothing parameters:
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.

To specify double exponential smoothing parameters:

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.

To specify Holt-Winters method parameters:

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.

To specify multiplicative decomposition parameters:

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.

To specify curve fitting parameters:

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.

To specify polynomial regression parameters:

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.

The stage needs to be set just so.

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.

To generate a summary of a forecast model’s performance metrics:
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).

Observing trends
You use trends to track in what direction the forecasted data moves.

To observe the trend of the historical data:
1. Open the forecast for which you want to create a trend.
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 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.
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.

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:

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. 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.

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.

Some examples of industries that experience cyclical demand are automobile, defense, mineral resources, and construction industries. To observe cyclicity within historical data:

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 Cyclicality toolbar. The Create Cyclicality 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 Cyclicality 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.

Editing chart properties

You use the Chart Properties wizard to edit the properties of forecast result charts.
Once you run a forecast model, you can customize the generated chart formatting to your specifications. To edit a forecast result chart:

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.

---

**Editing grid properties**

You use the Grid Properties wizard to edit the properties of forecast result grids.

Once you run a forecast model, you can customize the generated grid formatting to your specifications. To edit a forecast result grid:

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.
Chapter 7. Working with charts

Charts allow you to display your query results data in a visually rich format.

There are a variety of fully customizable charts, or display modes, that you can use to visually display your data. These charts can be formatted to suit your needs, and any number of charts can be created for a single result set. All of the charts that are created for a result set are saved with the result set, and managed through the Display editor. The Display editor contains three panes that help you manage your query result set display modes:

- Result set
- Layout structure
- Display modes

The Result set pane will display all of the available result set columns, regardless of the selected display mode. Query result columns will automatically be 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 that have been assigned aggregations. You can use the Result set pane to add grouping and aggregation 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 will only display the columns, groupings, and aggregations of the currently selected display mode. You can use the Layout structure pane to edit the individual layout structures of each of your available display modes.

The Display modes pane displays a list of all of the currently 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. Available display modes are:

- **Column Chart** - Data points are represented by vertical columns.
- **Bar Chart** - Data points are represented by horizontal columns.
- **Line Chart** - Data points are plotted on the x-y axes using inter-connected lines.
- **Area Chart** - 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 Chart** - Data points are displayed as wedges of a pie.
- **Scatter Chart** - Data points are plotted as discrete points, using a user-configurable data symbol.
- **Bubble Chart** - Data points are plotted using spherical data symbols, each of which can be sized according to an expression or column value.
- **Speedo Chart** - Data points are displayed as points on a dial.
- **Grid Chart** - Data points are displayed in a tabular grid.
Creating a chart from the query results grid

You can create a ready made chart from selections from the query results grid.

The fastest way to create a new chart is through the query results grid. This allows you to review the returned data before making a choice as to what to display, and how to display it. To create a new chart from the query results grid:

1. Navigate to the query that has the result set columns that you want to display in the Results editor.
2. Select the columns that you want to include in the chart by holding the Ctrl key and clicking each column.
3. 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 columns that contain character data, and value columns are 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.

4. Select each value column and select how it is aggregated from the Aggregation drop-down list.
5. Click Next. The Format page opens.
6. Specify the different formatting options of the chart and click Finish. The New chart wizard closes and the new chart is displayed in the Results editor.

Creating a chart from the Display editor

You can create a ready-made chart from selections from the Display editor.

To create a new chart from the Display editor, perform the following procedure:

1. Navigate to the query that has the result set columns that you want to display in the Display editor.
2. 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.
3. With the mouse button pressed, drag the columns to the Chart in the Display modes. Release the mouse button.
   A new chart appears in the Chart tree. This chart becomes default chart.
4. You can edit the chart by performing the following steps:
   a. Double-click the chart’s name or right-click and select Edit in the context menu.
   b. Specify the Chart type, Data Series and Chart Format
   c. Click OK to save modifications to the chart.

You have created a chart from the Display editor and optionally have made modifications to the chart.
Editing query result set columns

You use the Result set pane of the Display editor to modify the columns of a result set.

To modify result set columns:
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 context menu.
4. To roll up a column into the side group, right-click the column and select Side Group from the context menu.
5. To add aggregation to a column, right-click the column and select an aggregation type from the context menu. Available aggregations are:
   • Count
   • First
   • Last
   • Maximum
   • Minimum
   • Sum
   • Average
   • Standard Deviation
6. To remove aggregation from a column, right-click the column and select No Aggregation from the context 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 context 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.

To modify the layout structure of your query result set:
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 context menu. Available aggregations are:
   • Count
   • First
   • Last
   • Maximum
   • Minimum
- Sum
- Average
- Standard Deviation

6. To remove aggregation from a column, right-click the column and select **No Aggregation** from the context menu.

7. To remove a column from the layout structure, right-click the column and select **Remove** from the context 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.

---

### Managing display modes

You use the **Display modes** pane to add, edit, and delete the available display modes of the query result set.

Display modes let you visualize result set data in many different ways. You can create any number of display modes, each with its own content and visual formatting. Every result set starts with grid called **Results grid**. This is the standard query results grid that displays the first time you run a query. You can not edit this grid, but you can add a new one to the **Grid** node of the **Display modes** tree, and edit it’s layout as needed. To manage the display modes of a query result set:

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 context menu.

5. To edit a display mode, right-click it and select **Edit** from the context menu.

6. To delete a display mode, right-click it and select **Remove** from the context 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 context menu.

8. To set a given display mode as default, right-click it and select **Make Default** from the context menu. The default display mode will open first every time the query is run.

---

### Adding a display mode

You use the New chart wizard to add new display modes.

To add a new display mode:

1. Right-click either **Grid** or **Chart**, select **New**, and select the display mode that you want to add. The Data Series page of the New chart wizard opens.

2. Specify the columns that you want to display as category and value axes and click **Next**. The Format page opens.

3. Specify the different formatting options of the chart and click **Finish**. The New chart wizard closes and the new display mode is displayed in the **Display modes** pane.
Editing the chart type options of a display mode

Chart Type options determine the type of the chart and the visual format of the display mode.

To edit the chart type options of a display mode, perform the following procedure:
1. Open the Edit chart properties wizard using any one of the following methods:
   • In the Display modes pane, right-click a display mode and select Edit from the context menu.
   • Double-click a chart name
   • Click the toolbar button
2. Click the Chart Type tab. The Chart Type page opens.
3. Select the type of chart from the Chart type field.
4. Select the dimension of the chart from the Dimension drop-down list. Valid options are:
   • 2D - The chart is displayed as a flat, two dimensional object.
   • 2D With Depth - The chart is displayed as a two dimensional object, with data points displayed in three dimensions.
   • 3D - The entire chart is displayed in three dimensions.
5. If available, specify a Subtype.
6. Click OK. Your changes are saved and the Edit chart properties wizard closes.

Editing the data series options of a display mode

The Data Series options of a display mode determine what columns of the result set are used to generate the chart’s data points.

To edit the data series options of a display mode:
1. In the Display modes pane, right-click a display mode and select Edit from the context menu. The Edit chart properties wizard opens.
2. Click the Data Series tab. The Data Series page opens.
3. 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.
4. 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.
5. Select how the value column will be tallied in the Aggregation drop-down list.
6. To remove a column from the series tree, select the column and click Remove. The column is added back to the Result set tree.
7. Click OK. Your changes are saved and the Edit chart properties wizard closes.

Editing the chart format options of a display mode

The Chart Format options of a display mode determine the look and feel of the generated chart object.

To edit the chart formatting options of a display mode:
1. In the Display modes pane, right-click a display mode and select Edit from the context menu. The Edit chart properties wizard opens.
2. Click the Chart Format tab. The Chart Format page opens.
3. Specify the options under the **Series** node of the tree. **Series** options determine which category axis and value axis series are displayed in the chart, and how they are formatted.

4. Specify the options under the **Chart Area** node of the tree. **Chart Area** options determine the visual formatting of the chart object, such as the chart title, legend, and chart area color schemes.

5. Click **OK**. Your changes are saved and the Edit chart properties wizard closes.

### 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.

To specify **Series** chart format options:

1. Open the Chart Format page of the Edit display mode window.
2. Select **Series** in the tree. The **Series** options are displayed.
3. Specify whether the chart data points will be colored according to the value series or the category in the **Color By** drop-down list.
4. To include or exclude one of the value series in the chart, select or clear the check box in the **Visible** column.
5. Click **OK**. Your changes are saved and the Edit display mode window closes.

### Specifying Category chart format options

Category chart format options control how the chart data points are displayed along the category axis.

To specify **Category** chart format options:

1. Open the Chart Format page of the Edit display mode window.
2. Select **Category** in the tree. The **Category** options are displayed.
3. Specify what data series is used to sort the data points along the category axis in the **Order By** drop-down list.
4. If you specified **Values** in the **Order By** drop-down list, select the value column that you want to use.
5. Specify whether the data points along the category axis will be sorted ascending or descending in the **Order** drop-down list.
6. Select what type of cutoff that is used to restrict the display of data points along the category axis in the **Type** drop-down 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.
7. Specify the value column that you want to use to determine cutoff in the **Value Series** drop-down list.
8. Specify the value that you want to use to determine the cutoff in the **Value** field.
9. Select the **Visible** check box to make excluded data points visible.
10. Specify what label will be displayed with the excluded data points in the Label field.
11. Click OK. Your changes are saved and the Edit display mode window closes.

**Specifying Value(Y)Series chart format options**
Value(Y)Series chart format options control how the value series axis is displayed.

To specify Value(Y)Series chart format options:
1. Open the Chart Format page of the Edit display mode window.
2. Select Value(Y)Series in the tree. The Value(Y)Series options are displayed.
3. Specify the title of the value series axis in the Title field.
4. To make the title of the value series axis visible, select the Visible check box.
5. Specify the color of the value series data points in the Color field.
6. Expand the Value(Y)Series node and select Labels.
7. Specify the labels options of the values series axis.
8. Click OK. Your changes are saved and the Edit display mode window closes.

**Related tasks:**
"Specifying Labels chart format options” on page 171
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.

To specify Chart Area chart format options:
1. Open the Chart Format page of the Edit display mode window.
2. Select Chart Area in the tree. The Chart Area options are displayed.
3. Specify the background color of the chart area in the Background field.
4. To display an outline around the chart area, select the Visible check box.
5. 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.
6. 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.
7. Click OK. Your changes are saved and the Edit display mode window closes.

**Specifying Axes chart format options**
Axes chart format options control what axes of the chart are included, and how they are displayed.

To specify Axes chart format options:
1. Open the Chart Format page of the Edit display mode window.
2. Select Axes in the tree. The Axes options are displayed.
3. To include an axis in the chart, select the check box in the Visible column.
4. Specify the title of an axis in the Title column.
5. Select whether an axis will be linear or logarithmic in the Type column.
6. Expand the Axes node and select X-Axis.
7. Specify the X-Axis chart format options.
8. Specify the Y-Axis-1 chart format options.
9. Specify the Y-Axis-2 chart format options.
10. Click OK. Your changes are saved and the Edit display mode window closes.

**Specifying X-Axis chart format options**

X-Axis chart format options control how the x-axis of the chart is displayed.

To specify X-Axis chart format options:
1. Open the Chart Format page of the Edit display mode window.
2. Select X-Axis in the tree. The X-Axis options are displayed.
3. To display the x-axis of the chart, select the Visible check box.
4. To display the title of the x-axis, select the Visible check box next to the Title field.
5. Specify the title of the x-axis in the Title field.
6. Specify the font of the title in the Font field.
7. Specify the color of the x-axis in the Color field.
8. Select the position of the x-axis from the Origin drop-down 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.
9. 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.
10. Click OK. Your changes are saved and the Edit display mode window closes.

**Specifying Y-Axes chart format options**

Y-Axis chart format options control how the y-axis of the chart is displayed.

To specify Y-Axis chart format options:
1. Open the Chart Format page of the Edit display mode window.
2. Select Y-Axis in the tree. The Y-Axis options are displayed.
3. To display the y-axis of the chart, select the Visible check box.
4. To display the title of the y-axis, select the Visible check box next to the Title field.
5. Specify the title of the y-axis in the Title field.
6. Specify the font of the title in the Font field.
7. Specify the color of the y-axis in the Color field.
8. Specify the position of the y-axis in the Origin drop-down 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.
9. Specify whether the y-axis values will be displayed as a linear function or a logarithmic function in the Type drop-down list.
10. Click OK. Your changes are saved and the Edit display mode window closes.

Specifying Gridlines chart format options
Gridlines chart options control whether gridlines are displayed and how they are formatted.

To specify Gridlines chart format options:
1. Open the Chart Format page of the Edit display mode window.
2. 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.
7. Click OK. Your changes are saved and the Edit display mode window closes.

Specifying Labels chart format options
Labels chart format options control whether the labels of an axis are displayed and how they are formatted.

To specify Labels chart format options:
1. Open the Chart Format page of the Edit display mode window.
2. 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 drop-down list. Valid options include:
   • As is - The exact text of the column value 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 specified by you.

The following formats are available for date:
   • YYYYxMMxD
   • MMxDxYYYY
   • DDxMMxYYYY
   • YxMMxD
   • MMxDxYY
   • DDxMMxYY

The following formats are available for time:
The following formats are available for date:

- HHxMMxSS
- HHxMM

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. 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.

10. Click **OK**. Your changes are saved and the Edit display mode window closes.

**Specifying Range chart format options**

Range options allow you to manually format the range and step count of an axis of a chart.

To specify **Range** chart format options:

1. Open the Chart Format page of the Edit display mode window.
2. 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.
8. Click **OK**. Your changes are saved and the Edit display mode window closes.

**Specifying Title chart format options**

Title chart format options control whether the title of the chart is displayed and how it is formatted.

To specify **Title** chart format options:

1. Open the Chart Format page of the Edit display mode window.
2. Select **Title** in the tree. The **Title** options are displayed.
3. To display the title of the chart, select the **Visible** check box.
4. From the **Type** drop-down list, select whether the chart title is automatically generated using the axes columns or is custom text that you enter.
5. Specify the title of the chart in the **Text** field.
6. Specify the font of the title in the **Font** field.
7. Select the location of the chart from the **Location** drop-down list.
8. Click **OK**. Your changes are saved and the Edit display mode window closes.

**Specifying Legend chart format options**
Legend chart format options control whether a legend is displayed and how it is formatted.

To specify **Legend** chart format options:
1. Open the Chart Format page of the Edit display mode window.
2. Select **Legend** in the tree. The **Legend** options are displayed.
3. To display the legend of the chart, select the **Visible** check box.
4. To display the title of the legend, select the **Visible** check box that is next to the **Title** field.
5. Specify the title of the legend in the **Title** field.
6. Specify the font of the title in the **Font** field.
7. Specify the position of the legend in the **Position** drop-down list.
8. Specify the font of the text of the legend in the **Font Text** field.
9. To display an outline around the legend, select the **Visible** check box that is next to the **Outline** field.
10. Click **OK**. Your changes are saved and the Edit display mode window closes.

**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.

To specify **Plot Area** chart format options:
1. Open the Chart Format page of the Edit display mode window.
2. Select **Plot Area** in the tree. The **Plot Area** options are displayed.
3. Specify the background color of the plot area in the **Background** field.
4. To display an outline around the plot area, select the **Visible** check box that is next to the **Outline** field.
5. Specify the distance that you would like to inset the chart area from the edge of the plot area in the **Insets(Points)** fields.
6. Click **OK**. Your changes are saved and the Edit display mode window closes.

**Specifying Client Area chart format options**
Client Area chart format options control how the client area of the chart is displayed.

To specify **Client Area** chart format options:
1. Open the Chart Format page of the Edit display mode window.
2. Select **Client Area** in the tree. The **Client Area** options are displayed.
3. Specify the background color of the client area in the **Background** field.
4. To display an outline around the client area, select the **Visible** check box that is next to the **Outline** field.
5. Click **OK**. Your changes are saved and the Edit display mode window closes.
Chapter 8. Working with visual projects

Visual projects include visual reports and visual 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.

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 DataQuant for Workstation, DataQuant for WebSphere, or Lotus Notes® 8.

Visual Designer Environment

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, Layers and Output views which support the Visual Designer editor and assist in creating visual reports and dashboards.

Creating visual reports

Visual reports are page-based reports that include both text and graphics.

Using DataQuant 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 DataQuant for WebSphere, but you can not 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:

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, DataQuant 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 181](#) 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 182](#) 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.
Creating reports using the visual report wizard

The Create New Visual Report wizard will help you create a new visual report.

To create a new visual report using the Create New Visual Report wizard:

1. Open the Create New Visual Report wizard in one of the following ways:
   - 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.

To create a new visual report template:

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\DataQuant for Workstation\Templates\VR Templates\VisualReport1
   
   - For Windows Vista or later:
     
     \C:\Users\[UserName]\Application Data\IBM\DataQuant 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 DataQuant 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.


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.

You can open legacy reports directly or by importing the report. Once opened in DataQuant for Workstation V1.1 (or later), these reports can only be saved in a repository. They cannot be saved in a QMF catalog because once opened in V1.1 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.

To open a legacy visual report directly:
1. Select File > Open From. From the context 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 select. 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.

To import a legacy visual report:
1. From any perspective, select File > Import. The Import window 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 reports

You set up a connection information alias for each data source that will provide data for the visual report.

Visual reports are intended to be shared across the enterprise. In order to facilitate sharing and distributing visual reports, DataQuant for Workstation does not tie the queries that provide the data for the visual report to a specific data source. Instead it ties the query to a connection information alias.

When you add the queries that will be used in the visual report, you will 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:
1. You can open the Insert Connection window in one of the following ways:
   a. From the Project Explorer view, right-click the Connections node for the specific visual report. Select Insert Connection from the context menu. The Insert Connection window opens.
   b. 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 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 in one of the following ways:
- 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.

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.

Specifying visual report queries

Visual reports can include one or more queries that will be run to obtain the result data for the report.

You can create a new query that will be included in the report or you can use existing queries. All queries that you will use in the visual report must be defined in the Queries folder for that report. 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.

To specify a query that will be used in the visual report:

1. You can open the Insert Query window in one of the following ways:
   - From the Project Explorer view, right-click the Queries node. Select Insert Query from the context menu. The Insert Query window opens.
   - From the Project Explorer view, double-click the Queries node. The Insert Query window opens.
2. Specify a unique name for the query in the Query name field.
3. From the list of available connections, select the connection that points to the data source that you want to use for this query. To add a new connection information alias, click Add Connection. The Insert Connection window opens.
4. Select the data source you would like the connection to point to in the Data Sources tree. Specify a name for the connection in the Connection name field. Click Finish. The Insert Connection window closes and control returns to the Insert Query window.
5. Identify the query and specify where it resides in one of the following ways:
   a. If you are creating a new query, select Create a new query.
   b. 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 use the browse button to search for the query.
   c. 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 use the browse button to search for the query.
      Specify the query to be added in the Select data file box. You can use the browse button to search for the query.
6. 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.
7. You will automatically save the query when you save the visual project.

Note: You can also add a query by dragging the query from the Workspaces view to the Queries node of the visual report in the Project Explorer 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.

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:

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.

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:

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 context 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 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 design mode, select the **Design** tab.

**Managing report data retrieval**

You can control when data is retrieved for visual reports.

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:
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 context 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.

To assign data retrieval schedules to visual report queries:

1. Right-click the **Queries** node for the visual report in the **Project Explorer** view. Select **Assign Expiration Schedules** from the context 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.

To create an offline report from a visual report:

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.**

**Creating Visual dashboards**

A visual dashboard is a type of visual project that presents interactive or persistent data to a wide range of users.
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 DataQuant for Workstation, content developers can create a visual 
dashboard that can be viewed by multiple users with either DataQuant for 
Workstation, DataQuant for WebSphere, or Lotus Notes 8.

Note: You can view previously designed visual dashboards in DataQuant 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:

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
   • Queries
   • Globals
   • Scenes

4. Create data source connections.
   Visible Dashboards are intended to be shared across the enterprise. In order to 
facilitate sharing and distributing Visual Dashboards, DataQuant 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 Visual Project queries.

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 that you will use 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 you will elect 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 the 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 listbox objects) and data-driven layout objects (such as a grid or bar chart object) using the Connectivity tool from the Palette view.

    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 your dashboard**

Up front planning will facilitate developing a visual dashboard. As a content developer you must decide what to present, how much initial detail to provide, how to access more detailed information, and how to navigate to and from the scenes that make up your dashboard.

A storyboard is a plan for what your visual dashboard will include. You use a storyboard to design the basic features, functionality and presentation flow of the scenes and objects that will be included in your visual dashboard. A storyboard for a visual dashboard should specify the following:
There are various tools that you can use to create storyboards. You can sketch one using pencil and paper. 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. The following list of general topics is a guide to some of the design elements you should include in your plan:

**Displaying query results data**
You use layout objects or the List and Combo 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.

To create a new Visual Dashboard, perform the following steps:

1. Open the Create New Visual Dashboard wizard on one of the following ways:

- 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.

You have created a new Visual Dashboard using the Create New Visual Dashboard wizard.

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.

To create a new visual dashboard template:

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.
• 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.

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 \ DataQuant for Workstation\Templates\VD Templates\VisualDashboard1
   • For Windows Vista or later:
     C:\Users\[user_name]\Application Data\IBM \ DataQuant 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 DataQuant 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.

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:
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.

Visual dashboards are intended to be shared across the enterprise. In order to facilitate sharing and distributing visual dashboards, DataQuant 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:

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 context 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.

Note: The connection information alias is added with the name `ConnectionName`. You can edit the name of the connection information alias by right-clicking on the entry and selecting Rename.

Specifying visual dashboard queries

Visual dashboards can include one or more queries that will be run to obtain the result data that will be displayed in the dashboard.

You can create a new query that will be included in the dashboard or you can use an existing query. All queries that you will use in the visual dashboard must be defined in the Queries folder for that dashboard. You can add queries at any time while designing your dashboard. For each query that you include in the visual dashboard, you must specify what data source will be used. You do that by associating the query with a connection information alias.

To specify a query that will be used in the visual dashboard:

1. You can open the Insert Query window in one of the following ways:
   • From the Project Explorer view, right-click the Queries node. Select Insert Query from the context menu. The Insert Query window opens.
   • From the Project Explorer view, double-click the Queries node. The Insert Query window opens.

2. Specify a unique name for the query in the Query name field.
   The interface provides you with the capability to rename the query while designing the visual dashboard.
   To rename the query, perform the following steps from the Project Explorer view:
   a. Right-click the query that you want to rename and select Rename from the context menu.
      The query name is highlighted in the Project Explorer view.
   b. Type the new name for the query in the highlighted edit box.
   c. Click outside the edit box in the Project Explorer view.
      A message box opens and prompts whether you want to save the query.
   d. Click Yes to save the query.
      The query is renamed.

3. From the list of available connections, select the connection that points to the data source that you want to use for this query. To add a new connection information alias, click Add Connection. The Insert Connection window opens.

4. Select the data source you would like the connection to point to in the Data Sources tree. Specify a name for the connection in the Connection name field. Click Finish. The Insert Connection window closes and control returns to the Insert Query window.

5. Identify the query and specify where it resides in one of the following ways:
   a. If you are creating a new query, select Create a new query.
b. If the query exists, select **Attach to an existing query**. The **Select query** field becomes available. Specify the name of the query and where it resides in the **Select Query** field. You can use the browse button to search for the query.

c. 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 field. You can use the browse button to search for the query.

6. 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.

**Note:** You can also add a query by dragging a query from the **Workspaces** view to the **Queries** folder for the visual dashboard in the **Project Explorer** 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.

**Working with scenes**

A visual dashboard consists of one or more scenes that will display data to users.

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:

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 context 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 context 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 context 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.

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:
1. Open the Insert Parameter window in one of the following ways:
   • From the Project Explorer view, right-click the Locals node for the scene to which you want to add a parameter. Select New. Select Parameter from the context menu. The Insert Parameter window opens.
   • From the Project Explorer view, right-click the scene for which you want to create the parameter. Select New. Select Parameter from the context menu. The Insert Parameter window opens.
   • From the Project Explorer view, double-click the Parameters node for the scene to which you want to add a 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.

### 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.

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.

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:

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 context 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.

Setting up 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.

An example of using information zooming on a layout might be to design a carpool display with the locations of employees' homes, using a linear map layout object with three levels of detail. At the default zoom level (100%), the map displays markers to show the locations of each employee's home. At 200%, the map displays employee names. At 400%, the map displays employee names, phone numbers, and desired commute times. In this example, there are three levels of detail.
Setting up information zooming for layout objects:

When you set up information zooming for a layout you can provide more data from the same query as your user zooms in on a layout object.

All zoom levels in a layout object use the same query results data. For each layout object that you create, you can add multiple zoom levels. When you create a layout object a folder called Level1 is created. This folder contains all the elements that will be displayed whenever the object is viewed at its initial, 100% or lower zoom level.

To incorporate information zooming, you add additional levels to the layout object. Each level that you add is numbered sequentially and added as a folder in the Project Explorer view. You add the objects that you want to display at a particular zoom level to the level folder. Each sequential level is displayed as the user changes to a higher zoom level.

To add a level of detail to a layout object:
1. From the Project Explorer view, right-click the layout object’s DataTemplate folder to which you want to add an additional level of detail. Select Insert Level of Detail from the context menu.
2. An Insert Level of Detail message box is issued. This message asks how you want to initialize the new level. Select Yes to create a duplicate of the previous level. All the objects and their properties from the last level that appear in the folder are copied to the new level. Select No to create a blank level.
3. A folder for the new level is added to the layout object's folder in the Project Explorer. A new canvas labeled with the level name is presented in the editor. If you selected to model the new level on a previous level, all of the objects appear in the new level. If you selected to have a blank level, the canvas is empty.
4. After creating the new level, you can insert or delete objects, and edit the object properties. You can use any of the query results data that is available for the layout object.
5. When you create a new level, the zoom percentage that will trigger the level to be displayed is automatically assigned. The default value for each added level is twice the amount of the previous level. For example, the zoom level is 100% for the first level, 200% for the second level, and 400% for the third level. You can modify this value in one of two ways:
   - Edit the level's MinimumZoom property in the Properties view.
   - From the Project Explorer view, right-click the Data Template folder for the level whose zoom percentage you want to change. Select Edit Level Transitions from the context menu. In the Level of Detail Transition Editor dialog box, specify desired zoom levels for each level of detail transition point.

Note:

When you compose queries for layouts, include as many columns as you think might be valuable to end users. You can display only the most important or highest-level column information in the default level of detail, and then add a level of detail to reveal more column information.

Design your data template and levels of detail at the same zoom level as they will appear to the user. When you insert text into a data template or a scene, the font
size is set to scale to the zoom level. If you want the font size to remain constant regardless of the zoom level, change the **FixedSize** property for the **Font** object to **True** in the **Properties** view.

You can set up information zooming for a scene and for the layouts within the scene. In this case, if there are duplicate zoom levels, a scene level takes priority. For example, if your layout has the following four zoom levels; level 1 will display at the zoom percentage of 100%; level 2 will display at the zoom percentage of 125%; level 3 will display at the zoom percentage of 150%; level 4 will display at the zoom percentage of 200% and your scene has a level 2 that will display at 200%, then when your user switches to a 200% zoom level, the level 2 scene will be presented and not the level 4 object.

**Setting up information zooming for scenes:**

When you set up information zooming for a scene you can provide more data in different layouts and from multiple queries as your user zooms in on a scene.

For each scene that you create, you can add multiple zoom levels. When you create a scene a folder called **Level1** is created. This folder contains all the elements that will be displayed whenever the scene is viewed at its initial, 100% or lower zoom level.

To incorporate information zooming, you add additional levels to the scene. Each level that you add is numbered sequentially and added as a folder in the **Project Explorer** view. You add the objects that you want to display at a particular zoom level to the level folder. Each sequential level is displayed as the user changes to a higher zoom level.

For example, you might create a second level of detail in the scene and include all the same objects in as the first level of detail, but with a different query used for the layout.

To add a level of detail to a scene:

1. From the **Project Explorer** view, right-click the scene folder to which you want to add an additional level of detail. Select **New > Level of Detail** from the context menu.
2. An **Insert Level of Detail** message box is issued. This message asks how you want to initialize the new level. Select **Yes** to create a duplicate of the previous level. All the objects and their properties from the last level that appears in the folder are copied to the new level. Select **No** to create a blank level.
3. A folder for the new level is added to the scene's folder in the **Project Explorer** view. A new canvas labeled with the level name is presented in the editor. If you selected to model the new level on a previous level, all of the objects appear in the new level. If you selected to have a blank level, the canvas is empty.
4. After creating the new level, you can insert or delete objects, and edit the object properties.
5. When you create a new level, the zoom percentage that will trigger the level to be displayed is automatically assigned. The default value for each added level is twice the amount of the previous level. For example, the zoom level is 100% for the first level, 200% for the second level, and 400% for the third level. You can modify this value in one of two ways:
   - Edit the level's **MinimumZoom** property in the **Properties** view.
• From the **Project Explorer** view, right-click the **Scene** folder for the level whose zoom percentage you want to change. Select **Edit Level Transitions** from the context menu. In the Level of Detail Transition Editor dialog box, specify desired zoom levels for each level of detail transition point.

**Note:**

Design your scene and levels of detail at the same zoom level as they will appear to the user. When you insert text into a data template or a scene, the font size is set to scale to the zoom level. If you want the font size to remain constant regardless of the zoom level, change the **FixedSize** property for the Font object to True in the Properties view.

Create viewpoints to direct user attention to a particular level of detail. Using a jump to a viewpoint, you can allow users to go directly to the area in the scene and the level of the detail that contains desired information.

You can set up information zooming for a scene and for the layouts within the scene. In this case, if there are duplicate zoom levels, a scene level takes priority. For example, if your layout has the following four zoom levels; level 1 will display at the zoom percentage of 100%; level 2 will display at the zoom percentage of 125%; level 3 will display at the zoom percentage of 150%; level 4 will display at the zoom percentage of 200% and your scene has a level 2 that will display at 200%, then when your user switches to a 200% zoom level, the level 2 scene will be presented and not the level 4 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.

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:
1. Open the New Viewpoint window in one of the following ways:
   • In design 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 context 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.

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.

To modify a viewpoint:

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 context menu.

d. Press the Home key.

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.

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.

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.

- **BorderType.** 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.

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.
To insert a simple embedded scene object in your visual dashboard scene:

1. From the Project Explorer, open the source scene in design mode.
2. From the Palette view, click the EmbeddedScene object.
3. Click in the editor window to insert the embedded scene object. The EmbeddedScene 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 on 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 (unchecked) 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 EmbeddedScene 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.

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:

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 EmbeddedScene. The EmbeddedScene 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 on 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 EmbeddedScene 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.

All events are not available for assignment to every object. The **Events** view lists the events that are available for the object that is selected. 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:

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. Click an object in your dashboard (in either the scene editor or in the **Project Explorer** view). A list of events that can be assigned to the selected object is displayed in the **Events** view. All events are not available for assignment to every object.

3. From the **Events** view, double-click the event that you want to assign to the current object. The [objectname] Behavior window opens.

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 have been 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 listed.

5. Select an action from the list of available actions. You are prompted for additional information depending on the action that you choose. Depending on the type of object to which you are assigning the event, one or more of the following actions are available:
   - Jump to new location
   - Set values
   - Navigate to embedded scene
   - Navigate back
   - Execute procedure command
   - Execute SQL statement
   - Print scene
   - Execute shell command
   - Show message
   - Execute JavaScript
   - Export to Excel
   - Set focus
   - Reload scene
   - End the session
   - Export to PDF
   - Send e-mail
   - Send event to embedded scene
   - Invalidate query cache
   - Refresh component
   - Add Launch LOB event
• Open URL

6. The event action that you define is listed in the Action cell of the list box. For each action you can also specify a condition. 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.

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. The event to which you have added one or more actions appears in bold in 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 Dashboard:

Use the Jump to new location event action to assign and define jump events to objects in your Visual Dashboard. At runtime, jump events take the user from an object in the Visual Dashboard to another scene or viewpoint of a scene or to a visual report.

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.

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 dashboard 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 Dashboard, perform the following steps:

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 from the list of available actions.

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 dashboard or to another dashboard or visual report. Select This dashboard to specify that the destination for the jump will be to a scene or viewpoint within the current dashboard. Select Another visual project to specify that the destination for the jump will be another visual dashboard or a visual report.

6. If you selected This dashboard, 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 dashboard in the Visual project field. Click the ellipses button to search for and open a dashboard 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 dashboard 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.


11. 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.

12. 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.

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:

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 from the list of available actions.

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 following objects:

- scene objects

You can incorporate ExternalEvent events in scene objects to trigger any of the following event actions:

- Jump to new location
- Navigate to embedded scene

**Note:** The **Navigate to embedded scene** event action is only available to objects other than the EmbeddedScene object if an EmbeddedScene object is in the current scene.

- Execute SQL statement

**Note:** The **Execute SQL statement** event 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 the session
- Export to PDF
- Send e-mail
- Send event to embedded scene
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 **EmbeddedScene** object.

Embedded scenes are added and set up automatically when a user clicks directly on 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 **EmbeddedScene** 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 **EmbeddedScene** object.
- To specify an event (other than the **Click** event) to trigger entering the embedded scene of an **EmbeddedScene** object.
  
  For example, you might want a **DblClick** event to trigger entering the embedded scene of an **EmbeddedScene** object.

To add and define a **Navigate to embedded scene** event action to an object:

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 **Navigate to embedded scene** from the **Action** list box.
3. Click **Next**. A window requesting the embedded scene action parameters opens.
4. 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.
5. 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.
6. Click **Finish**. The **Navigate to embedded scene** event action is added. The Add New Action window closes.
7. 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.
To add a **Navigate back** event action to an object:

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** from the **Action** list box.
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.

To add an **Execute SQL statement** event action to an object:

1. Select the object in the editor window. The [objectname] Behavior window opens. Click the **Add** icon. The Add New Action window opens.
   From the **Events** view, double-click the event that you want to trigger the action of running the SQL statement.
2. Select **Execute SQL statement** from the **Action** list box.
3. 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:
   
   ```sql
   Update MyTable Set Shipped = 1
   Where CustomerID = [value of g_nCustID global variable]
   ```

   You would enter the following text in the SQL Statement field:
   ```sql
   "Update MyTable Set Shipped = 1 Where CustomerID = " + g_nCustID
   ```
4. 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.
5. 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.
6. 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 / application 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.

7. Click Finish. The **Execute SQL statement** 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.

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.

To add an **Execute procedure** event action to an object:

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 **Execute procedure** from the **Action** list box.

3. Use the **From Repository** field to enter the procedure statement name and location.

4. Select the **Hide procedure result windows** check box to hide the results window after running the procedure statement.

5. Select the **Refresh display on completion** check box to refresh the dashboard display after running the procedure statement.

6. 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.

7. 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.

8. Click Finish. The **Execute procedure** 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.

*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.

To add an **Execute JavaScript** event action to an object:

1. Select the object in the editor window. The [objectname] Behavior window opens. Click the **Add** icon. The Add New Action window opens. From the **Events** view, double-click the event that you want to trigger the action of running JavaScript.

2. Select **Execute JavaScript** from the **Action** list box and click **Next**. The Execute JavaScript action parameters page opens.

3. Type JavaScript commands in the **JavaScript** field and click **Finish**. The JavaScript command action is added. The Add New Action window closes.

4. From the [objectname] Behavior window, click **Apply** to apply all actions that have been added for the event.

5. Click **OK** to close the [objectname] Behavior window.

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.

To incorporate a **Print scene** event action in an object:

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** from the list of available actions.

4. Click **Next**.
   - The Print scene action parameters page opens.
   - If, when printed, the scene does not fit within the print area, select the **Shrink to page** check box.

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.

To add a **Send email** event action to an object:

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** from the list of available actions.

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.

    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.

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:

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 from the list of available actions.
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.
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:

<table>
<thead>
<tr>
<th>Action</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExportToPDF</td>
<td>=ExternalEventID =’Btn1ExportToPDF’</td>
</tr>
<tr>
<td>Action</td>
<td>Condition</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>SendEmail</td>
<td>ExternalEventID='Btn2SendEmail'</td>
</tr>
</tbody>
</table>

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.

For example, you can run a shell command to open Notepad or an e-mail program.

To add an **Execute shell command** event action to an object:
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** from the list of available actions.
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.
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 Show message event action:

Use the **Show message** event action to display a message.

To add a **Show message** event action to an object:
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** from the list of available actions and click **Next**. 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 will be issued 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 DataQuant for Workstation.
6. Select the severity level of the message from the **Logging level** drop-down list. Your choices are:
   - error
7. Click Finish. The Show message action is added. The Add New Action dialog closes.

8. From the [objectname] Behavior dialog, click Apply to apply all actions that have been added for the event.

9. Click OK to close the [objectname] Behavior dialog.

Adding an Export to Excel event action:

Use the Export to Excel action to export a specified event to Excel.

See system requirements in the Installing and Managing guide for information about the version of Excel required to support the Export to Excel feature.

To add an Export to Excel event action to an object:

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 from the Action list box and click Next. The Export to Excel action parameters page opens.

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 in 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**: This opens Save As dialog. Enter the name of the file in the File name field to save it.

   - **Save to the temp file**: This 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 in the Name field.

     - Enter the parameter's description in the Description field.

     - Select the Has default value check box and specify the full file path in the text field; for example, C:\DOCUME~1\<user>\LOCALS~1\Temp\<software_product_name>\reporter<user>\pdf\..

   - **Save to the specified file**: This automatically saves the generated file in a specified root directory when the event occurs.
8. If you select **Save to the specified 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 DataQuant 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 DataQuant 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.

9. 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 the temp file** or **Save to the specified file**, the **Result options** section is active.

10. 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**.

   **Note:** Valid parameter types are **FilePath**, **Literal**, and **Text**. The Insert Parameter window opens.

11. Enter a name for the parameter in the **Name** field.

12. Enter the parameter’s description in the **Description** field.

13. Select the **Has default value** check box and specify the full file path in the text field.

14. Click **Finish**. The **Export to Excel** action is added. The [objectname] Behavior window is updated.

15. 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 DataQuant so that objects generated from various DataQuant actions are exported to a specific location on either the local file system or 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.

To add a **Set focus** event action to an object:

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** from the list of available actions.
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.

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.

To add a Reload scene event action to an object:

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 from the list of available actions.


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.

By adding an End Session event action to a dashboard object, you can close the application from the dashboard 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:

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 from the list of available actions.

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.

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.

To add an Export to PDF event action to an object:

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 from the list of available actions.
4. Click Next. An Export to PDF specific window opens.
5. In the Export type radio group, select one of three radio buttons:
   - **Open**: This opens Save As dialog. Enter the name of the file in the File name field to save it.
   - **Save to the temp file**: This 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 in the Name field.
     - Enter the parameter's description in the Description field.
     - Select the Has default value check box and specify the full file path in the text field; for example, C:\DOCUME~1\<user>\LOCALS~1\Temp\<software_product_name>\reporter-<user>\pdf\..
   - **Save to the specified file**: This automatically saves the generated file in a specified root directory when the event occurs.
6. If you select Save to the specified 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 DataQuant 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 DataQuant 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 the temp file or Save to the specified 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**.

**Note:** Valid parameter types are **FilePath**, **Literal**, and **Text**. The Insert Parameter window opens.

9. Enter a name for the parameter in the **Name** field.

10. Enter the parameter's description in the **Description** field.

11. Select the **Has default value** check box and specify the full file path in the text field.

12. Click **Finish**. The Insert Parameter window closes and control returns to the Add New Action window.

13. Select the parameter that you want to use from the **Result parameter** drop-down lists and click **OK**. The Add New Action dialog closes.

14. From the [objectname] Behavior window, click **Apply** to apply all actions that have been added for the event.

15. 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 DataQuant so that objects generated from various DataQuant actions are exported to a specific location on either the local file system or server file system.

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.

The **Refresh Component** event action is appropriate when querying related components only.

To add a **Refresh component** event action to an object:

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 Action** from the list of available actions.

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.

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.

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:
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** from the list of available actions.
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.

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.

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:
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** from the list of available actions.
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.

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 DataQuant for Workstation, the data opens in the associated system application. In DataQuant for WebSphere, the data opens in the web browser.

*Adding an Open URL event action:*

Use the **Open URL** event action to allow users to open hyperlinks from visual dashboards.

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:
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** from the list of available actions. Click **Next**.
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 DataQuant 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.
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.

**Protecting visual dashboard data**

You can compile read-only versions of visual dashboards so that other developers can use the dashboards, but are prevented from accessing and modifying the underlying data objects.

To protect visual dashboard data:
1. Open the previously saved visual 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 Results as**.
3. Select a mode from the **Project renderer** list.
   The mode determines how the compiled visual dashboard is rendered.
4. Click **Finish**.
5. From the Save [Object] wizard, select whether to save the visual project to a file or repository and click **Next**.
6. Provide the necessary information on the Save to File or Save to Repository wizard and click **Finish**.

You have compiled a read-only version of your visual project. The data associated with the visual dashboard is protected and cannot be modified by other users.

**Queries and visual projects**

Visual projects (visual reports, dashboards, and applications) can include one or more queries to run to obtain the result data.

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 visual report queries**

Visual reports can include one or more queries that will be run to obtain the result data for the report.

You can create a new query that will be included in the report or you can use existing queries. All queries that you will use in the visual report must be defined in the **Queries** folder for that report. 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.

To specify a query that will be used in the visual report:
1. You can open the Insert Query window in one of the following ways:
   - From the **Project Explorer** view, right-click the **Queries** node. Select **Insert Query** from the context menu. The Insert Query window opens.
• From the **Project Explorer** view, double-click the **Queries** node. The Insert Query window opens.

2. Specify a unique name for the query in the **Query name** field.

3. From the list of available connections, select the connection that points to the data source that you want to use for this query. To add a new connection information alias, click **Add Connection**. The Insert Connection window opens.

4. Select the data source you would like the connection to point to in the **Data Sources** tree. Specify a name for the connection in the **Connection name** field. Click **Finish** The Insert Connection window closes and control returns to the Insert Query window.

5. Identify the query and specify where it resides in one of the following ways:
   a. If you are creating a new query, select **Create a new query** .
   b. 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 use the browse button to search for the query.
   c. 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 use the browse button to search for the query.
         Specify the query to be added in the **Select data file** text box. You can use the browse button to search for the query.

6. 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.

7. You will automatically save the query when you save the visual project.

**Note:** You can also add a query by dragging the query from the **Workspaces** view to the **Queries** node of the visual report in the **Project Explorer** 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.

---

## Working with the Visual Designer

In DataQuant for Workstation, the **Visual Designer** perspective presents an editor window, several views and the menus that you will use to create visual projects.

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 you will design and create your visual project. From the editor window you can switch between design mode and runtime mode. The runtime mode of the editor will display your project as it will be seen by a user.

- **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.

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.

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.
Working with the editor

The Visual Designer editor window displays a canvas on which you design your visual project.

From the editor window you can switch between design mode and runtime or preview mode. The runtime or preview mode of the editor displays your project as a user will see it.

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 Visual Designer editor can present your visual project in both runtime and edit modes depending on how you choose to open the project. The tabs at the bottom of the window allow you to toggle between the modes.

- **Design mode**
  You use Design mode to add content to your visual project. When you add objects they are represented in the editor by placeholder objects.

- **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 dashboard as the end-user will see it.

- **Print Preview mode**
  You use Print Preview mode to test the content in your visual project. In Print 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 Print Preview mode of the editor will display your project as a user will see it.

You can control how your visual project will be rendered in the Print Preview and Runtime modes by using the rendering mode drop-down list in the workstation menu bar.

You can select from the following rendering mode options:

- **Rich Client** optimizes the content to be viewed in a rich client viewer.
- **HTML** optimizes the content to be viewed in a web browser.
- **Optimized HTML** optimizes the content to be viewed in a web browser with a single image with map and several controls above the image.

Using Optimized HTML rendering improves performance over standard HTML because Optimized HTML does not create controls for every component on the scene and it does not use XSLT transformation. However, there are some limitations with the Optimized HTML rendering mode. The limitations are as follows:

- Every control on a scene is located above a non-control component.
- Conditional Embedded Scene printing is not supported.

For example, you cannot print part of a scene image.
– Some controls cannot get and set focus.
  For example, if you use Optimized HTML, you cannot get and set focus for the Slider control.
– Some controls cannot be zoomed.
  For example, the check box and radio button controls cannot be zoomed when rendered in Optimized HTML.
– Wait cursor does not exist.
• Flash optimizes the content to be viewed in a Flash player.
• PDF optimizes the content to be printed.

**Rulers, guides, and the grid**

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.

**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.

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 properties for groups of objects**

You can specify properties for groups of objects that you have selected in the editor window.

To specify property values for a group of objects:

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 desired 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.

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:

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.

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**:

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 on a function to insert it into the property value field. Double click on 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 on the property or function. To insert content, position the cursor in the property field and then double click on 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:

<table>
<thead>
<tr>
<th>Task</th>
<th>Rule</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display a calculated value, such as a parameter or column name</td>
<td>Use equals sign (=) for calculated expressions</td>
<td>=cust_num</td>
</tr>
<tr>
<td>Display a literal value</td>
<td>Type the literal exactly as you want it to appear in text</td>
<td>Sales Organization</td>
</tr>
<tr>
<td>Display a text string in an expression</td>
<td>If the string is part of a combined expression that also includes calculated values, columns or parameters, it must be surrounded by quotation marks (&quot; &quot;)</td>
<td>=&quot;My name is: &quot;+fname</td>
</tr>
<tr>
<td>Concatenate two or more text strings</td>
<td>Use the plus sign (+) to concatenate text strings</td>
<td>=Click here to go to&quot;+Dept+&quot; department</td>
</tr>
<tr>
<td>Add two or more numeric values</td>
<td>Use the plus sign (+) to add constants or parameters of numeric data</td>
<td>=Param1+5</td>
</tr>
<tr>
<td>Insert a carriage return in an expression</td>
<td>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</td>
<td>=city+newline+state</td>
</tr>
<tr>
<td>Display the value of another property</td>
<td>Use the equals sign (=) plus the object name and the property name with dot notation</td>
<td>=Text1.Text</td>
</tr>
</tbody>
</table>

**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:

```javascript
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:

```javascript
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.

```javascript
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.

Each scene object (or any graphic object within a scene) has a set of events that designers can employ. These events can be configured to respond to end-user actions (such as a mouse click, mouse movement, or keystroke combinations) in very specific ways. Additionally, designers can set event triggers to coincide with processing activities. For example, a designer might want to invoke a specific action to coincide with a LoadComplete event.

An object cannot make use of an event until the event is assigned to the object by the designer.
As an user navigates through the visual project, the events assigned to an object trigger predefined event actions, such as jumping to a new scene, setting a parameter value, executing SQL statements, or launching other applications.

Not all events are available for assignment to every object. The events that are available for the selected object are listed in the Events view from Visual designer perspective.

When assigning an event to an object, designers can specify additional actions to be triggered when the event occurs. See Setting up event action navigation for information about how to assign events to objects and how to specify the actions that are triggered when the event occurs.

**Working with the Palette view**

The Palette view displays all of the graphic objects that are available for insertion into your visual project.

The objects are grouped into the following palettes (also referred to as drawers):

- **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 listbox 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.

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:

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 context menu. Depending on how you want to view the objects in the Palette view, select one of the commands that are offered in the context menu.

**Modifying the Palette view display settings**

You can vary the way objects are displayed in the Palette view.

To vary the way objects are displayed in the Palette view:

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 object. For example, a Visual Dashboard, Visual Report or Query object.

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 context 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 context 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 context 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.

To change an object's default property value:

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 desired.
4. When you are finished changing properties, return to the editor window, right-click the object and select **Set as Default** from the context 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 context 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** or **ListBox** objects) and data-driven layout objects (such as a **Grid** or **BarChart** object).

You can also use the **Connectivity** tool to define dependencies between non-query driven objects such as **TextBox** 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:

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 **ListBox** or **Combo** 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.

You can find the Timer object in the Miscellaneous palette.

To insert a Timer object:
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 **EmbeddedScene** object if the **EmbeddedScene** 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 Java script
   - Export to Excel
   - Set focus
   - Reload scene
   - End session
   - Export to PDF
   - Send Email
   - Send Event to Embedded Scene

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.

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.
- **MultiPolyline**
  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.

MultiPolygon
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.

To insert text into a visual project:
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.

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 theLineStyle 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 multipolylines in your visual project.

To insert a line, arrow, double arrow, polyline or multipolyline object into your visual project:
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.

- **DoubleArrow**: You use this object to draw a single line with an arrowhead on the starting point and ending point. The DoubleArrow 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.

- **MultiPolyline**: You use this object to draw multiple polylines. Select the MultiPolyline 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 Multipolyline object.

3. From the Properties view, modify the properties, such as color, width, and style to draw your line as desired. 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.

To insert a shape object into your visual project:

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.

   - **RoundRectangle**: You use this object to draw a rectangle that has rounded corners. Select the RoundRectangle 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.

- **MultiPolygon**: You use this object to draw multiple polygons. Select the *MultiPolygon* 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 *MultiPolygon* object.

3. From the *Properties* view, modify the properties, such as color, width, and style to draw your shape object as desired. 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.

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:

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 theGlobals 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.

**Inserting an alignment panel object:**

The AlignmentPanel object is an invisible reference object that you can use to group and position child objects relative to a single point.

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:

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 AlignmentPanel object and insert it in the editor window. The AlignmentPanel 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 AlignmentPanel 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.

Note:
- In a layout object, the Location property of the AlignmentPanel 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.

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:

TextBox
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.

CheckBox
An object that presents two choices to the user for selection. A check box control can be either checked or unchecked.

RadioGroup
A composite object containing a set of radio buttons that present clickable choices to the user.
HorzSlider
A window that is used as a scrolling control for selecting an integer value within a range of values.

VertSlider
A window that is used as a scrolling control for selecting an integer value within a range of values.

Combo
An object that combines an editable text field with a list box. When displayed, the Combo object displays a box with the list of choices associated with the particular field. The user may enter additional choices using the editable text field of the Combo object.

ListBox
An object that displays to the user a list of possible values that can be selected for a particular field.

DateTime
An object that makes it easy for users to enter date or time information which can then be used by other objects.

Inserting a control object:

Control objects are graphical objects that are used to display information or accept user input.

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:

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 ListBox 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 a DateTime control object:

A DateTime 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.

To insert a DateTime object into a visual dashboard and set up a connection to another object:
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 DateTime object from the Controls palette and insert it in the editor window.
3. Specify whether the DateTime object will accept a date or a time. To specify that the object will accept date values, select from the Property view the value Date for the Mode property value. To specify that the object will accept time values, select the value Time for the Mode property value.
4. Insert the object whose behavior will be controlled by the date or time entered in the DateTime object. You can insert an object that will display the return value of the DateTime object or you can insert an object that will use the DateTime return value as a parameter.
5. To display the DateTime 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 DateTime object whose return value you want to display. Double-click the Value property.
   e. The LabelObject.Text property will equal the DateTimeObject.Value.
6. To use the DateTime 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 context 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 DateTime. Set a default value for the parameter. Click OK.
   c. Set the query parameter value to equal the return value of the DateTime 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 DateTime 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 DateTime object Value property.
You can also use the Connectivity tool to set up the connection between the DateTime object and the Grid object.

Inserting a Slicer object:

A Slicer object is used to filter the multidimensional data that is displayed in a layout object.

To insert a Slicer object into a visual dashboard and set up a connection to another layout object:
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.

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.

Adding data symbol objects

A Data Symbol object is an icon that is used to represent a data point in a layout object.

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:

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.

VerticalValueBar
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.
VerticalLabelBar
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.

HorizontalValueBar
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.

HorizontalLabelBar
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.

PriceIntervalBar
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.

EventBand
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.

ThresholdBand
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.

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:
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.

**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.

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.

To insert a connector into a visual project:
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 desired.
4. Click **Runtime** or **Preview** to view the results.
Tip:
- To force all connection lines to connect at a single specified point, insert a ConnectionPoint and position it as desired.
- 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.

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.

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:
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 desired.
3. When you are finished, return to the editor window, right-click the object and select Add to Custom Palette from the context menu. The object retains the properties that you set and is placed on the Custom palette.

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 **EmbeddedScene** object and the **EmbeddedContent** object.

An **EmbeddedContent** object is a container that provides the means to add external content to a visual report or dashboard. You view the embedded content in the embedded content object. An **EmbeddedScene** object is a container that provides the means to navigate from one scene to another. The embedded scene actually displays in the embedded scene object.

### Inserting an embedded scene object:

An **EmbeddedScene** object is a container through which you can view all or a portion of another scene and then navigate to that scene.

You can use the **EmbeddedScene** 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:

1. From the **Project Explorer** view, open the source scene in design mode.
2. From the **Palette** view, click the **EmbeddedScene** object.
3. Click in the editor window to insert the **EmbeddedScene** object. The EmbeddedScene 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 (unchecked) 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 EmbeddedScene 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 EmbeddedScene object appears in the folder for your scene.

Inserting an embedded content object:

An EmbeddedContent 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.

To insert an EmbeddedContent object in your visual dashboard scene:
1. From the Project Explorer view, open the source scene in design mode.
2. From the Palette view, click the EmbeddedContent object.
3. Click in the editor window to insert the EmbeddedContent object. The EmbeddedContent object is drawn based on default size properties. Click in the editor window and drag the cursor to insert the EmbeddedContent object drawn to your specific size. The Embedded content wizard opens.
4. Specify where to find the content for the object in the Content location field. You can use the Browse button to search for the location.
5. 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.
6. Click OK. An EmbeddedContent object is created. An entry is made for the object in the Project Explorer view.

Note: Embedded content objects can not display HTML in the Flash rendering runtime mode. If you intend on including an embedded content object in a visual dashboard that will be displayed using the Flash rendering mode, make sure you select Text in the Content type drop-down list of the EmbeddedContent Wizard. Alternatively, you can display HTML content from a Flash rendered dashboard by navigating to a scene from a separate dashboard that is not rendered in Flash. For more information, see “Assigning jump events to objects in your Visual Dashboard” on page 205

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.

The following chart layouts are available:
- BarChart
- PieChart
- ColumnChart
- ScatterChart
- XYChart
- StockChart
- CandlestickChart
- TimelineChart
- EventBandChart
- MultivariateChart
- LinearMap
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

**Inserting a table:**

Table objects let you display query results in a simple table.

To insert a Table object:

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.

To insert a simple Table object:

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.

To insert a populated Table object:

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 drop-down 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 in Flash rendering mode only.
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. Click Next. The Specify paging options page opens.
15. Select the Enable paging check box to enable paging. Paging allows you to display only a certain amount of rows at a time.
16. Specify the amount of rows that you want to display at a time in the Rows per page list.
17. Select the first row in the result set to display in the First displayed row list.
18. Select the Add navigation bar check box to automatically create page forward and page backwards buttons that control paging actions.
19. 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.

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:

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.

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:

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 bar or column chart:**

Bar charts and column charts display data as a series of bars on an x or y axis.

To insert a bar or column chart:

1. Double-click the BarChart or ColumnChart object in the Layouts palette of the Palette view. The BarChart Wizard or ColumnChart 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.

To insert an extended bar or column chart:

1. Double-click the **BarChart** or **ColumnChart** object in the **Layouts** palette of the **Palette** view. The BarChart Wizard or ColumnChart 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.

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:

1. Double-click the BarChart or ColumnChart object in the Layouts palette of the Palette view. The BarChart Wizard or ColumnChart 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.

To insert a pie chart using a relational query:

1. Double-click the PieChart object in the Layouts palette of the Palette view. The PieChart 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 PieChart 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.

To insert an extended pie chart:

1. Double-click the **PieChart** object in the **Layouts** palette of the **Palette** view. The PieChart 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 PieChart 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.

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:
1. Double-click the **PieChart** object in the **Layouts** palette of the **Palette** view. The PieChart 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 your 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 PieChart 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.

To insert a multivariate chart:

1. Double-click the **MultiVariateChart** object in the **Layouts** palette of the **Palette** view. The MultiVariateChart 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 MultiVariateChart 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.

To insert a scatter or XY chart:

1. Double-click the **ScatterChart** or **XYChart** object in the **Layouts** palette of the **Palette** view. The ScatterChart Wizard or XYChart 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.

To insert a stock chart:

1. Double-click the **StockChart** object in the **Layouts** palette of the **Palette** view. The StockChart 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 stocks 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 StockChart 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.

To insert a candlestick chart:

1. Double-click the CandlestickChart object in the Layouts palette of the Palette view. The CandlestickChart 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 stock's 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 CandlestickChart Wizard closes and the chart is displayed in the editor window.
Inserting a timeline chart:

Timeline charts display a time series as a curve.

To insert a timeline chart:
1. Double-click the **TimelineChart** object in the **Layouts** palette of the **Palette** view. The TimelineChart 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 TimelineChart 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.

To Insert an event band chart:
1. Double-click the **EventBandChart** object in the **Layouts** palette of the **Palette** view. The EventBandChart 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 EventBandChart 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.

To insert a cluster graph:
1. Double-click the **ClusterGraph** object in the **Layouts** palette of the **Palette** view. The ClusterGraph 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 ClusterGraph Wizard closes and the graph is displayed in the editor window.

**Inserting a linear map:**

Linear maps display spatial data in a geographic context.

To insert a linear map:
1. Double-click the **LinearMap** object in the **Layouts** palette of the **Palette** view. The LinearMap 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 LinearMap Wizard closes, and the map displays in the editor window.

**Inserting a matrix:**

Matrices display data in a two-dimensional array of cells.

To insert a matrix:
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 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.

To insert an organization chart:

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 OrganizationChart 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.

To insert a tree chart:

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 TreeChart Wizard closes and the chart is displayed in the editor window.

**Inserting a dial:**

Dials display points of data as a range between two numbers on a metered scale.

To insert a dial:

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.

**Changing a layout object's query:**

You can change a layout object’s query by selecting **Change Query** in the **Project Explorer** menu.

To change a layout's object's query:

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 context 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 object in the Parameters folder of the layout object’s data template.

To specify a query parameter value:
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.

**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.

To change a layout object’s default data symbol:
1. Double-click on 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.

**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.

To change a layout object’s default connector:
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.

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.

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:
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

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.

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:
- CandlestickChart
- EventBandChart
- LinearMap
- ScatterChart
- StockChart
- TimelineChart
- ColumnChart
- BarChart
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.

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:
1. In the Project Explorer view, right-click the data template folder that you want to copy. Select Copy from the context 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 context 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.

Creating a new data template:

You can use the Insert Data template command to add a new data template to a layout object.

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:
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 context 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.
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 wish to 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. For example, in the organization chart described above, you want to add a column chart that shows a sales team member's quarterly sales totals underneath their name. You open the data template folder in the editor window and add a parameterized ColumnChart object underneath 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.

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.

**Modifying a data template’s visibility:**

You can specify whether a data template layer is displayed or not displayed in the layout object.

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:

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**.

**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.

**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.

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:

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.

**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.

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:

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 desired layer from the context 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 desired 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.

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.

Java Script Modules
Java Script 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 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.

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.

To create a color map:
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 context 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.

To use a color map:

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.

**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.

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.

To create a new color sequence:

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 context menu. The New Color Sequence wizard opens.
   - From the Project Explorer view, click theGlobals 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.

To use a color sequence:
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.

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.

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.

To create a new stock image:
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 context 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 a Picture object in your visual project.

You can associate a stock image to a 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 a picture object and insert the picture object 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.

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.

DataQuant for Workstation provides the following global parameters:
- **ExternalEventID** is a global parameter, set to a particular control on a scene. It saves EventID from **Send to Embedded Scene** 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.

To add a user-defined global parameter:

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 context 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.

To use global parameters when specifying object properties:
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.

**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.

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:
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 context 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.

**Working with translation tables**

Translation tables enable visual designers to publish a single visual dashboard that can be viewed in multiple languages.

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.

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 context 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.Text10.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 context 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 DataQuant 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.

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:

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 desired shape in the editor window. The ListBox Wizard
   opens.

3. Select No and click Next. The Fill in option and value list page opens.
4. Click **Finish**. The ListBox 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 desired 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 **ListBox1 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.
Chapter 9. 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 DataQuant 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 DataQuant procedure command
- comment lines
- blank lines
- RUN commands that run other procedures or queries
- substitution variables

Note:
- DataQuant 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 DataQuant for Workstation command line.
- DataQuant supports procedures with logic (REXX procedures). Procedures with logic include REXX language directives. Users must have OBJECTREXX installed to run procedures with logic.

Creating a new procedure

You can use the Procedure window to create new procedures.

To create a 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. Enter the commands that will be included in the procedure in the Procedure editor. 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 DataQuant procedure commands in the procedure with logic, the DataQuant procedure commands must be enclosed in single quotes.
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 DataQuant job scheduler or your local scheduler.

**Scheduling procedures to run**

You can schedule your procedure to run at any given time by using the operating system scheduler or the DataQuant for WebSphere scheduler. Procedures that have been scheduled to run at another time are referred to as scheduled jobs.

The local scheduler uses the specific to Windows operating system scheduler to run a job. The DataQuant for WebSphere scheduler actually uses DataQuant for WebSphere and does not depend on the operating system you use.

You must have a procedure saved in a repository, QMF catalog, or file system (for local scheduler only).

To schedule a procedure to run:

1. Open the Schedule New Job wizard by selecting **File > New > Scheduled Job**.
2. On the first page of the wizard, select what type of scheduler to use. The available options are:
   - **Use local scheduler** to schedule a job using your local scheduler.
   - **Use DataQuant for WebSphere scheduler** to schedule a job using the DataQuant for WebSphere scheduler.
   Click Next.
3. On the **Select job parameters** page of the Schedule New Job wizard, specify a job name.
4. From the Repository connection list, select the repository connection that will be used to connect to the repository required for running the procedure.
   - If you are using DataQuant for Workstation and scheduling a job by using your local scheduler, the list is populated with all the repository connections that are available to your current version of DataQuant for Workstation.
   - If you are using DataQuant for Workstation and scheduling a job by using the DataQuant for WebSphere scheduler, the list is populated with all the web service connections that are available to your current version of DataQuant for Workstation.

   **Note:** You do not have to be currently connected to the selected repository.

5. Click **Set User Information** to specify the user ID and password that will be used to connect to the repository.
6. Specify which data source to use. For this, select one of the following options:
   - **Use data source defined in the procedure** to specify that the data source against which the procedure must be run is defined within the procedure.
   - **Select data source from the list** to select the data source against which the procedure must be run. You can select from the list of data sources that are available in the repository. Click **Set User Information** to specify the user ID and password that will be used to connect to the data source.

7. Specify which procedure to run. Select one of the following options:
   - **From File** to specify that the procedure is located in a file system. You must specify the full path to the file. You can use the ellipsis (...) to browse for the file.
• **From Repository** to specify that the procedure is located in a repository. You can use the ellipsis (...) to open the Open From Repository window. From this window, expand folders until you expose the repository with the required procedure. Select the repository and click **OK**.

• **From QMF Catalog** to specify that the procedure is located in a QMF catalog. You can use the ellipsis (...) to open the Open From QMF Catalog window. From this window, specify the data source with the required QMF catalog. In addition, you specify the owner and name of the procedure object. Click **From List** to open the Object List window. From this window you can search the QMF catalog and list the objects that it contains. Select **OK**.

8. Specify variables required by the procedure that must be run. Click **Add variables**. Set the name of the variable in the Name cell. Set the value for the variable in the Value cell. This table is pre-populated with any variables that are included in the procedure. If a procedure calls another procedure, and the second level procedure calls for variables, the second level procedure variables are not in the list and you cannot define values for these variables when the procedure is run as a scheduled job. Procedures that call second level procedures that require variable values must not be scheduled. In addition, procedures that depend on user interaction for their variable values must not be scheduled.

9. Click **Schedule**. If you are using a local scheduler, the operating system scheduler window opens. If you are using the DataQuant for WebSphere scheduler, the Schedule Job window opens. You can specify when to start the job and how frequently it must be performed. For the DataQuant for WebSphere scheduler, available frequency options are:
   • Hourly
   • Daily
   • Weekly
   • Monthly
   • Once

10. You can create one or more schedules for a job. The **Schedules** field lists each of the schedules that have been created for the job. To create a new schedule, click **Create**. The new schedule is automatically named. The fields necessary to define a schedule become available.

11. When you have completed creating the schedules, click **OK**. The calendar page of the wizard closes. Control returns to the Specify parameters of the task page of the wizard.

12. Click **Finish**. The job is scheduled to run. You can view a list of scheduled jobs by selecting **View > Scheduled Jobs**.

### Viewing scheduled jobs

You can view a list of all the jobs that have been scheduled to run.

To view a list of scheduled jobs:

1. Select **View > Scheduled Jobs**. The Scheduled Jobs window opens.

2. The Scheduled Jobs window has one or both of the following two pages:
   • If you are viewing scheduled jobs using the DataQuant for WebSphere application interface, the dialog includes the Local Jobs page. This page lists the jobs that have been scheduled to run using this version of DataQuant for WebSphere. Each scheduled job is listed by **Job Name**, **Next Run Time**, **Last Run Time**, **Last Finish Time**, and **Last Result**. **Last Result** displays whether
the job ran successfully. If Last Result displays a zero, the job ran successfully. If Last Result displays a one, the job did not run successfully.

- If you are viewing scheduled jobs using the DataQuant for Workstation application interface, the dialog includes the Local Jobs page. This page lists the jobs that have been scheduled to run using your local scheduler. Each scheduled job is listed by Job Name and Next Run Time.
- If you are viewing scheduled jobs using the DataQuant for Workstation application interface, the dialog includes the Remote Jobs page. This page lists the jobs that have been scheduled to run using DataQuant for WebSphere. Jobs are listed by Job Name, Next Run Time, Last Run Time, Last Finish Time, and Last Result. Last Result displays whether the job ran successfully. If Last Result displays a zero, the job ran successfully. If Last Result displays a one, the job did not run successfully. In addition, the web service repository connection that will be used to connect to DataQuant for WebSphere is listed in the Repository connection field.

3. You can delete a scheduled job from the list by selecting the job from the list and clicking the Remove Job icon. You will receive a confirmation message.

4. You can add a scheduled job to the list by clicking the Add Job icon. The Schedule New Job window opens. You will specify the necessary information to schedule the job and click OK. You will return to the list of scheduled jobs. The new job will be listed.

5. You can edit the parameters of a scheduled job by selecting the job from the list and clicking the Edit Job icon. The Schedule New Job window opens. You will make the changes to the parameters and click OK to close the window and return to the list of scheduled jobs.

6. You can export the scheduled job. Click the Export Job icon. Specify the location where the scheduled job will be saved.

7. You can import a scheduled job. Click the Import Job icon. Navigate to the location where the scheduled job is saved.

**Restoring scheduled jobs after logon information change**

You must change the password for the existing scheduled jobs that are run using the DataQuant for WebSphere scheduler, if the repository storage or repository password of the user who created the scheduled jobs is changed.

**Note:** The user who created scheduled jobs must change the password for them to ensure that these scheduled jobs will remain available for this user.

To change the password for scheduled jobs:

1. From the main menu, select View > Scheduled Jobs to open the Scheduled Jobs dialog.

2. From the Repository connection list, select the connection for which you want to change the password.

3. In the dialog that opens, specify the valid repository password and click OK to close the dialog. The actual repository password that you specified is automatically applied to all scheduled jobs that use the current repository connection.

**Tip:** If you want to specify different logon information for a separate scheduled job, select this job from the list, click Edit Job, and specify the repository login and password on the "Specify job parameters" page.
Converting legacy job definition files

Job definition files created using QMF for Windows must first be migrated to a new format before they can be run using DataQuant for Workstation.

To migrate legacy job definition files to a new format:
1. Select File > Import. The Import wizard opens.
3. Specify the directory where the legacy job definition files (.jdf) can be found in the Job files directory. You can use the Browse button to search for the directory.
4. Select the job files that you want to migrate to the new format from the Job files list. Each .jdf file that resides in the directory is listed in the Job files list.
5. Click Next. The .jdf files will be prepared for migration. The Server mapping page of the wizard opens.
6. Before the migration to the new format can be completed you must identify the data sources that will be used in the jobs. In QMF for Windows, data sources were referred to as servers and were defined in server definition files (SDF). Each server that is used in the legacy job file is listed in the QMF Server cell of the Server mapping table as it appeared in the SDF. These servers must be configured in the repository as data sources. This is usually done by the DataQuant for Workstation administrator. In the Repository Data Source cell of the Server mapping table specify the data source name for the server as it was configured in the repository.
7. To specify the name and password that will be used to login to the data source, select the data source and click Set User Information. The Set User Information window opens. Specify the name and password. Click OK. The information is listed in the Repository Data Source cell of the Server mapping table.
8. Click Next and specify the directory where the new job definition files (.jdfx) will be saved in the Output directory field. You can use the Browse button to search for the directory.
9. After you complete the process of migrating the job file to the new format you can run the job in one of the following ways:
   • Schedule the DataQuant for Workstation job file (.jdfx) as a task using your operating system’s scheduler. For more information on scheduling tasks, see the documentation for your scheduler.
   • Run the job file (.jdfx) by using the DataQuant for Workstation command library interface. For more information, see Running jobs from the command library interface in API help.

Procedure commands

Each procedure command is described below. Descriptions include syntax diagrams and usage examples.

Interface differences

Each DataQuant 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

Provides a summary of the syntax rules that are common to each command.

Typically, each line of a procedure contains a single command. However, you can continue 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 following the hyphens through the end of the line is a comment and is not interpreted when running the procedure.

You can use substitution variables to parameterize a procedure. Each time 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 a RUN PROC command, from global variables, or by displaying the Enter Substitution Variables window.

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 below, you must select at least one from the list.

>>-- DISPLAY --- QUERY ------------------<<
  +-- PROC ----+
  +-- FORM ----+
  +-- REPORT --+

Optional Parameters

When a parameter is shown below the main path, it is optional. When all parameters are shown in a list below 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 DataQuant 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 DataQuant 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:

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. The procedure above could also be written:

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 above is being run by someone who is signed on as ADMIN, then it could also be written:

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 in the first example.

**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:

```sql
RUN QUERY ADMIN."Test Case for Query @ HQ"
```

**Objects stored in Repository folders**

When DataQuant 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:

```sql
RUN QUERY TESTQ
```

Notice that the syntax of the example that is listed above matches the syntax of the third example in the QMF Catalog section. Therefore when a procedure command is submitted there is a search order that DataQuant uses to locate the objects referenced in the procedure.

1. DataQuant 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 above, 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. DataQuant 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. DataQuant 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. DataQuant 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>` as above, 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:

```sql
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**

```plaintext
>>>-- BOTTOM -------------------<<
```

**Example**

BOTTOM

**CONNECT command**

The **CONNECT** command will establish 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**

```plaintext
>>>-- CONNECT ---------------- TO ServerName ----------------------------<<
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerName</td>
<td>The name of a database server. If the name includes spaces, enclose it in quotes.</td>
</tr>
<tr>
<td>UserName</td>
<td>The user name that will be used to connect to the database server. UserName is an optional parameter.</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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 diagram format) 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:

```plaintext
>--- CONVERT --------- QueryName ----------------------------><
  + QUERY +++
  (+ SUBSTITUTE = YES/NO ++)
```

To convert a query currently in an open window (in temporary storage):

```plaintext
>--- CONVERT QUERY ----------------------------><
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryName</td>
<td>The name of the query that is to be converted.</td>
</tr>
<tr>
<td>SUBSTITUTE</td>
<td>Defines whether substitution should take place for query variables. Available options are:</td>
</tr>
<tr>
<td></td>
<td>• YES. If you have variables in your query, QMF attempts to substitute values for them. If all the variables are defined (either with the &amp;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.</td>
</tr>
<tr>
<td></td>
<td>• NO. No variable names in your query are resolved.</td>
</tr>
</tbody>
</table>

**Example**

CONVERT QUERY
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:

```plaintext
>--- DISPLAY --------------- ObjectName ----------------------------><
  + QUERY +++
  (+ &Variable = Value +++
  + FORM +++
  + PROC +++
  + TABLE +++
  + REPORT +++
  + DASHBOARD-+
```

To display an object derived from the content in the last open window:

```plaintext
>--- DISPLAY --------------- ObjectName ----------------------------><
  + QUERY +++
  + FORM +++
  +.MAIN---------
  +.BREAK1-------
  +.BREAK2-------
  +.BREAK3-------
  +.BREAK4-------
  +.BREAK5-------
  +.BREAK6-------
  +.CALC--------
```
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` 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 to place it within quotes, as is shown in the example above. 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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>The name or the full object key of the object to be displayed.</td>
</tr>
<tr>
<td>&amp;&amp;Variable</td>
<td>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 <code>DISPLAY</code> command. If there are variables in the query or procedure that are not given values on the <code>DISPLAY</code> 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.</td>
</tr>
</tbody>
</table>

**Examples**

```
DISPLAY USER1.QUERY1
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 +)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TableName</td>
<td>The name of the table for which to create a query.</td>
</tr>
<tr>
<td>TYPE</td>
<td>Specifies the type of SQL query to create. The default is SELECT.</td>
</tr>
<tr>
<td>IDENTIFIER</td>
<td>The correlation name to be associated with the table in the resulting query.</td>
</tr>
<tr>
<td></td>
<td>It is ignored when TYPE=INSERT. There is no default.</td>
</tr>
</tbody>
</table>

**Examples**

```
DRAW Q.STAFF (TYPE=SELECT)
```

**EDIT command**

The **EDIT** command will display 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:

```
>>-- EDIT ------------------------ObjectName -------------------------------<<
  (+- QUERY ----+  (+- &Variable = Value ----+)
    +- FORM ----+
    +- PROC ----+
    +- TABLE ----+
```

To edit an object derived from the content in the last open window:

```
>>-- EDIT ------------------+ QUERY --------------------------<<
  +- FORM ----+
  +- PROC ----+
  +- REPORT -+
```

In the **EDIT** command, you can provide either the object name or the full object key of the object being edited. If the procedure is stored in a QMF catalog, objects stored in the same catalog can be edited using only their object name. For example, if the Q.STAFF query is stored in the same catalog as the procedure, you would use **EDIT Q.STAFF** to edit the Q.STAFF query.

If the procedure is stored in the repository, objects stored along side the procedure in the same repository folder can be edited 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 **EDIT Q.STAFF** to edit the Q.STAFF query.

However, if the object to be edited is not stored in the procedure, nor in the same repository folder, then you must supply the full object key in the **EDIT** command. For example, **EDIT "qmf:/.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 to place it within quotes, as is shown in the example above. To display the current object in visual report format, you must use the **SHOW** command.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>The name or the full object key of the object to be edited.</td>
</tr>
<tr>
<td>&amp;Variable</td>
<td>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 <strong>EDIT</strong> command. If there are variables in the query or procedure that are not given values on the <strong>EDIT</strong> 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.</td>
</tr>
</tbody>
</table>

**Examples**

**EDIT USER1.QUERY1**

**EDIT "qmf:/.workspaces/Traditional DBA View/Query1"**

**EDIT QUERY**

**ERASE command**

The **ERASE** command removes an object (query, form, procedure, or table) from the database.

**Syntax**

`>> -- ERASE --ObjectName-- <<
  -- QUERY +
  -- (CONFIRM = YES/NO) +
  -- FORM +
  -- PROC +
  -- TABLE +
`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>The name of the object to remove from the database.</td>
</tr>
<tr>
<td>CONFIRM</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Example**

**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 DataQuant for WebSphere.

`>>-- Execute CommandLine --<<`
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommandLine</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Example**

EXECUTE c:\programs\notepad.exe

**EXPORT command**

The EXPORT command saves objects that reside on the database or are currently in an open window to a file.

**Syntax**

To export an object from a database to a file:

```plaintext
>>-- EXPORT --- QUERY --- ObjectName TO FileName ------------------------------------------<
+-- PROC --+              (+ CONFIRM = YES/NO -+  
+-- SAVETIME = YES/NO -+  

>>-- EXPORT FORM ObjectName TO FileName ------------------------------------------<
  (+ CONFIRM = YES/NO -+  
  +-- LANGUAGE = ENGLISH/SESSION -+  
  +-- SAVETIME = YES/NO -+  

>>-- EXPORT TABLE ObjectName TO FileName ------------------------------------------<
  (+ CONFIRM = YES/NO -------------------------------+  
  +-- DATAFORMAT = CSV/DBF/HTML/IXF/PDF/SHP/TEXT/WQML/XLS/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 -----------------------------+  
  +-- SAVETIME = YES/NO -----------------------------+  

Note: EXPORT TABLE always exports names.

To export an object currently in an open window to a file:

```plaintext
>>-- EXPORT --- QUERY --- TO FileName ------------------------------------------<
+-- PROC --+              (+ CONFIRM = YES/NO -+  
+-- SAVETIME = YES/NO -+  

>>-- EXPORT FORM TO FileName ------------------------------------------<
  (+ CONFIRM = YES/NO -+  
  +-- LANGUAGE = ENGLISH/SESSION -+  
  +-- SAVETIME = YES/NO -+  

>>-- EXPORT DATA TO FileName ------------------------------------------<
  (+ CONFIRM = YES/NO -------------------------------+  
  +-- DATAFORMAT = CSV/DBF/HTML/IXF/PDF/SHP/TEXT/WQML/XLS/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 -----------------------+  
```

296 Getting Started with DataQuant
Parameter | Description
--- | ---
ObjectName | The name of the object to export from the database.
FileName | The name of the file to which the object will be exported.
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 | 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:
- zero (0) for text format
- two (2) for HTML format
- three (3) for CSV format
  Note: The separator is always a comma unless the locale uses decimal comma. In this case, a separator of semi-colon is used.
- 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.
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.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATEFORMAT</td>
<td>Specifies how the date is formatted in the HTML, CSV or TXT export file. 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. Text can be quoted using single quotes (') to avoid interpretation. <strong>Note:</strong> 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.</td>
</tr>
<tr>
<td>TIMEFORMAT</td>
<td>Specifies how the time is formatted in the HTML, CSV or TXT export file. 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. Text can be quoted using single quotes (') to avoid interpretation. <strong>Note:</strong> 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.</td>
</tr>
<tr>
<td>OUTPUTMODE</td>
<td>Specifies the output format of the data.</td>
</tr>
<tr>
<td>LOBSINFILE</td>
<td>Specifies whether or not LOBs are included in the exported data.</td>
</tr>
<tr>
<td>LOBSTO</td>
<td>The location to save the LOBs.</td>
</tr>
<tr>
<td>LOBFILE</td>
<td>The base name of the exported LOBs.</td>
</tr>
<tr>
<td>CCSID</td>
<td>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.</td>
</tr>
<tr>
<td>COLUMNHEADINGS</td>
<td>Specifies whether or not to export column headers. It is only available for export to HTML, CSV or TEXT files.</td>
</tr>
<tr>
<td>UNICODE</td>
<td>Specifies whether graphic columns will be saved as UNICODE. This option is only applicable when saving data in DXF format.</td>
</tr>
<tr>
<td>SPLIT</td>
<td>Specifies whether the report will be split into pages. Use only when exporting visual reports.</td>
</tr>
<tr>
<td>ORDER</td>
<td>Controls how the report will be split into pages. Use only when exporting visual reports.</td>
</tr>
<tr>
<td>* 0 - each visual report page will be exported into separate files.</td>
<td></td>
</tr>
<tr>
<td>* 1 - a single output file contains visual report pages in &quot;across then down&quot; order.</td>
<td></td>
</tr>
<tr>
<td>* 2 - a single output file contains visual report pages in &quot;down then across&quot; order.</td>
<td></td>
</tr>
<tr>
<td>WIDTH</td>
<td>Specifies the width in units for a report page. <strong>Note:</strong> Used for the both classic and visual reports.</td>
</tr>
<tr>
<td>LENGTH</td>
<td>Specifies the length in units for a report page. <strong>Note:</strong> Used for the both classic and visual reports.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| UNITS       | Specifies the measurement units that are used in the WIDTH and LENGTH parameters.  
**Note:** Use only when exporting visual reports.                                                                                                                                                                                                                                                                                                                                                         |
| USEFORMPS   | Specifies whether or not to use the Windows form associated with the object.                                                                                                                                                                                                                                                                                                                                                               |
| SEPARATOR   | Specifies a string that will be used as a page separator. Use quotes if necessary.  
**Note:** Use only when exporting visual reports.                                                                                                                                                                                                                                                                                                                                                           |
| MODE        | Specifies whether the query result data is saved with formatting and added calculated columns. You can specify one of the following values:  
- **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.  
  This is the default value for PDF format.  
  **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.  
- **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 for all formats other than PDF.  
  **Note:** MODE RAW always exports names. This also applies when the MODE parameter is omitted.                                                                                                                                                                                                                                                                                       |
| SAVEATSERVER| Specifies whether or not to include the Root output directory in the export path of an object. This directory is set on the Server-Side File System page of the Preferences window. In DataQuant 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.                                                                                                           |

**Example**

```plaintext
EXPORT QUERY USER1.QUERY TO C:\Queries\query1.qry
```

The following is an example of using DATEFORMAT and TIMEFORMAT parameters:

```plaintext
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:

```plaintext
EXPORT QUERY USER1.QUERY TO query1.qry (SAVEATSERVER=YES
```

The result of the command will be a new file in C:\Temp\query1.qry

**Related tasks:**
- Setting the directory location preference for objects generated by Export actions
- You can set preferences in DataQuant so that objects generated from various DataQuant actions are exported to a specific location on either the local file system or server file system.

---

Chapter 9. Working with procedures 299
FORWARD command

The FORWARD command scrolls forward in 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 currently 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 +)
   (+- 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 +)
---```

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; +)
---```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>The name of the object to import into the database.</td>
</tr>
<tr>
<td>FileName</td>
<td>The name of the file from which the object will be imported.</td>
</tr>
<tr>
<td>CONFIRM</td>
<td>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.</td>
</tr>
<tr>
<td>COMMENT</td>
<td>Specifies a comment with the imported object. Enclose the text of the comment in quotes.</td>
</tr>
<tr>
<td>SHARE</td>
<td>Specifies whether other users are allowed to use the imported object.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>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 DSQEC_FORM_LANG global variable.</td>
</tr>
<tr>
<td>ACTION</td>
<td>Specifies whether the entire database table will be replaced or the new data will be appended to the existing table.</td>
</tr>
<tr>
<td>LOBSFROM</td>
<td>The location to save the LOBs.</td>
</tr>
</tbody>
</table>

**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 Enter Substitution Variable Values window when you run the object.

Any new local variables created are valid only for the current query, report, or dashboard.

**Syntax**

```javascript
>>>-- LIMIT LOCAL ( VariableName = Value, ... ------<<
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VariableName</td>
<td>The name of the local variable to set available values.</td>
</tr>
<tr>
<td>Value</td>
<td>The value or list of values which will be available to select from in the Enter Substitution Variable Values window</td>
</tr>
</tbody>
</table>

**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 reside on a database or in an open window (temporary storage).

**Syntax**

To send an object that resides in a database:

```bash
>>>-- 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 ------------------<<
+ TABLE +
+ ( --- 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 ++

MAIL ------------ ObjectName TO Address ------------------<<
+ 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 ++
+ DATEFORMAT = Java date format string ++
+ TIMEFORMAT = Java time format string ++
+ METHOD = SPLIT/CONT -------------------++

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 ++
+ DATEFORMAT = Java date format string ++
+ TIMEFORMAT = Java time format string ++
+ METHOD = SPLIT/CONT -------------------++

To send an object that currently resides in an open window (in temporary storage):

MAIL --- QUERY --- TO Address ------------------<<
+ PROC ++
+ ( --- 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 ++

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 ++
Parameter | Description
---|---
ObjectName | The name of the object to send.
FROM | The email of the sender.
CCLIST | The email addresses of the recipients.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATAFORMAT</td>
<td>Specifies the file format of the attached data object. If you omit this</td>
</tr>
<tr>
<td></td>
<td>parameter, the DSQQW_EXP_DT_FRMT global variable supplies the format to be</td>
</tr>
<tr>
<td></td>
<td>used. You can specify in DSQQW_EXP_DT_FRMT global variable:</td>
</tr>
<tr>
<td></td>
<td>• zero (0) for TEXT format</td>
</tr>
<tr>
<td></td>
<td>• two (2) for HTML format</td>
</tr>
<tr>
<td></td>
<td>• three (3) for CSV format</td>
</tr>
<tr>
<td></td>
<td>• four (4) for IXF format</td>
</tr>
<tr>
<td></td>
<td>• five (5) for dbase III files</td>
</tr>
<tr>
<td></td>
<td>• six (6) for Shapefile format</td>
</tr>
<tr>
<td></td>
<td>• seven (7) for WQML format</td>
</tr>
<tr>
<td></td>
<td>• eight (8) for XML format</td>
</tr>
<tr>
<td></td>
<td>• nine (9) for PDF format</td>
</tr>
<tr>
<td></td>
<td>• ten (10) for XLS format</td>
</tr>
<tr>
<td></td>
<td>If you specify IXF, the DSQQW_EXP_OUT_MDE global variable can be set to</td>
</tr>
<tr>
<td></td>
<td>either zero (0) for character mode System/370 IXF or one (1) for PC/IXF.</td>
</tr>
<tr>
<td></td>
<td>When exporting reports, HTML format adds the &lt;PRE&gt; tag, TEXT is for a</td>
</tr>
<tr>
<td></td>
<td>text-based file, and PDF is for a PDF-based file.</td>
</tr>
<tr>
<td>SUBJECT</td>
<td>The email subject line reference.</td>
</tr>
<tr>
<td>BODY</td>
<td>The contents of the email message.</td>
</tr>
<tr>
<td>FORMAT</td>
<td>The email format. Supported formats are Text and HTML.</td>
</tr>
<tr>
<td>SMTPSERVER</td>
<td>The name of a predefined SMTP server or address of a custom SMTP server.</td>
</tr>
<tr>
<td>SMTPPORT</td>
<td>The SMTP server port number.</td>
</tr>
<tr>
<td>SMTPUSER</td>
<td>The user name that is used to authorize on SMTP server.</td>
</tr>
<tr>
<td>SMTPPASSWORD</td>
<td>The password that is used to authorize on SMTP server.</td>
</tr>
<tr>
<td>ATTACHMENT</td>
<td>The name and path to the files that will be attached.</td>
</tr>
<tr>
<td>TYPE</td>
<td>The type that specifies the exported files are visual reports.</td>
</tr>
<tr>
<td>METHOD</td>
<td>The method separated pages.</td>
</tr>
<tr>
<td>DATETIMEFORMAT</td>
<td>Specifies how the date is formatted in the HTML, CSV or TXT export file.</td>
</tr>
<tr>
<td></td>
<td>Date formats are specified by Java date pattern strings. Within date</td>
</tr>
<tr>
<td></td>
<td>pattern strings, unquoted letters from ‘A’ to ‘Z’ and from ‘a’ to ‘z’ are</td>
</tr>
<tr>
<td></td>
<td>interpreted as pattern letters representing the components of a date</td>
</tr>
<tr>
<td></td>
<td>string.</td>
</tr>
<tr>
<td></td>
<td>Text can be quoted using single quotes (’) to avoid interpretation.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If the format string includes spaces, enclose it in quotes. For</td>
</tr>
<tr>
<td></td>
<td>more information about Java format strings, see the Java 2 SDK, Standard</td>
</tr>
<tr>
<td></td>
<td>Edition Documentation.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TIMEFORMAT</td>
<td>Specifies how the time is formatted in the HTML, CSV or TXT export file. 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. Text can be quoted using single quotes (') to avoid interpretation. <strong>Note:</strong> 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.</td>
</tr>
</tbody>
</table>

**Example**

MAIL QUERY TO abc123@mail.com

The following is an example of using DATEFORMAT and TIMEFORMAT parameters:

MAIL DATA TO someone@example.com (SUBJECT=Test, SMTPSERVER=smtp.example.com, DATEFORMAT="yyyy MM dd", TIMEFORMAT=HH:mm

**PRINT command**

The PRINT command prints a copy of an object that is in the current window (temporary storage) or resides on a database.

**Syntax**

**Note:** This command is ignored in DataQuant for WebSphere.

To print a copy of an object that resides on a database:

```plaintext
>>> PRINT --------- ObjectName ------------------------------------<<
  +- QUERY +
  +- (PRINTER = Printer +
    + TABLE +

To print a copy of the current object in an open window (temporary storage):

```plaintext
>>> PRINT --------- QUERY --------------------------------------<<
  +- PROC +
  +- (PRINTER = Printer +

```plaintext

```plaintext
>>> PRINT REPORT ---------------------------------------------<<
  (  +- DATETIME = YES/NO ------------------------------------+
      +- PAGENO = YES/NO -------------------------------------+
      +- PRINTER = Printer -------------------------------------+
      +- FORM = FORM/formname------------------------------+
      +- COPIES = NumCopies----------------------------------+
      +- TYPEFACE = Fontname -------------------------------+
      +- SIZE = FontSize -------------------------------------+
      +- BOLD = YES/NO ---------------------------------------+
      +- ITALIC = YES/NO --------------------------------------+
      +- CHARSET = ANSI/DEFAULTS/SYMBOL/SHIFTJIS/GB2312/ -+
          HANGUEL/CHINESEBIGS/GEM/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 -----------------------------+
```
## Parameter | Description
--- | ---
ObjectName | The name of the object to print.
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.
FORM | Specifies the form to use when generating the report. You can specify the keyword FORM to use the current form object, or specify the name of a form saved in the database.
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 DSQW_RPT_LEN_TYP and DSQW_RPT_NUM_LNS global variables are used.
WIDTH | 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.
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
PRINT QUERY
PRINT REPORT (LENGTH=60, DATETIME=NO

RESET GLOBAL
The RESET GLOBAL command deletes the names and values of global variables that have been set using the SET GLOBAL command.

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 or queries that reside on the database or are current objects in an open window (reside in temporary storage).

Syntax
To run an object that resides on the database:
>>>- RUN ------------ ObjectName ----------------<-
   " QUERY " ( "&&Variable = Value ----+
   " CONFIRM = YES/NO ---------------+
   " FORM = FORM/formname------------+
   " ROWLIMIT = NumRows--------------+

>>>- RUN ------------ ObjectName -------------------<-
   " PROC " ( "&&Variable =

To run an object from an open window (temporary storage):
>>>- RUN QUERY ---------------------------------------------------<-
 ( "&&Variable = Value ----+
 ( " CONFIRM = YES/NO ---------------+
 ( " FORM = FORM/formname ------------+
 ( " ROWLIMIT = NumRows/NO -----------+

>>>- RUN PROC ------------------------------------------------------<-
 ( "&&Variable = Value ...

Chapter 9. Working with procedures 307
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>The name of the query or procedure to run.</td>
</tr>
<tr>
<td>&amp;&amp;Variable</td>
<td>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 RUN command. If there are variables in the query or procedure 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.</td>
</tr>
<tr>
<td>CONFIRM</td>
<td>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.</td>
</tr>
<tr>
<td>FORM</td>
<td>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.</td>
</tr>
<tr>
<td>ROWLIMIT</td>
<td>Specifies the maximum number of rows to retrieve for the query.</td>
</tr>
</tbody>
</table>

**Example**

RUN QUERY USER1.QUERY1 (&&Var1=10)

**SAVE command**

The SAVE command stores the contents of an object in temporary storage into the database.

**Syntax**

```plaintext
>>> SAVE --- QUERY --- AS ObjectName -------------------------------<<
    (+ PROC ++
    { (+ COMMENT = text ---+
      + CONFIRM = YES/NO ++
      + SHARE = YES/NO ++
    
    >>> SAVE FORM AS ObjectName -------------------------------------<<
    (+ COMMENT = text ------------+ 
    + CONFIRM = YES/NO -----------
    + SHARE = YES/NO -----------
    + LANGUAGE = ENGLISH/SESSION +

    >>> SAVE DATA AS TableName ----------------------------------------------------<<
    (+ COMMENT = text ----------------------------------------------------------+
    + CONFIRM = YES/NO ---------------------------------------------------------
    + METHOD = REGULAR/Fast/Fastsafe-------------------------------------------+
    + SCOPE = NumChars --------------------------------------------------------
    + ROWIDADD = YES/NO -------------------------------------------------------
    + ROWIDNAME = text ---------------------------------------------------------
    + ROWIDISP = ALWAYS/BYDEFAULT -------------------------------------------+
    + ROWIDISPn = EXCLUDE/CONVERT/always/ByDefault--
    + ACTION = REPLACE/APPEND -----------------------------------------------+
    + SPACE = database.tablespace---------------------------------------------+
    + MODE = GRID/RAW ---------------------------------------------------------+
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>The name to assign to the object (query, form, procedure, or table) when it is saved.</td>
</tr>
<tr>
<td>COMMENT</td>
<td>Specifies a comment with the saved object. Enclose the text in quotes.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>CONFIRM</td>
<td>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.</td>
</tr>
<tr>
<td>SHARE</td>
<td>Specifies whether other users are allowed to use the saved object.</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>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.</td>
</tr>
<tr>
<td>METHOD</td>
<td>Specifies the method of saving the data.</td>
</tr>
<tr>
<td>SCOPE</td>
<td>Specifies the commit scope of the data.</td>
</tr>
<tr>
<td>ROWIDADD</td>
<td>Specifies whether to add the Row ID column to the table.</td>
</tr>
<tr>
<td>ROWIDNAME</td>
<td>Specifies the name to assign to the new Row ID column.</td>
</tr>
<tr>
<td>ROWIDDISP</td>
<td>Specifies the disposition of the new Row ID column.</td>
</tr>
<tr>
<td>ROWIDDISPnnn</td>
<td>Specifies the disposition of existing Row ID columns.</td>
</tr>
<tr>
<td>ACTION</td>
<td>Specifies whether to replace the entire database table or append data to the existing table.</td>
</tr>
<tr>
<td>SPACE</td>
<td>Specifies the name of the table space in the database that will be used to save the data. The table space name that you specify must match the default table space name that was set up for your user ID in the Save Data resource limit Default table space. If you have been given permission to override the default table space name, you can specify any name in this field. Permission to override the table space name is specified using the Default table space can be overridden option, located in the Save Data tab of the resource limits settings.</td>
</tr>
<tr>
<td>MODE</td>
<td>Specifies whether the query result data is saved with formatting and added calculated columns. You can specify one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
</tbody>
</table>

**Example**

```
SAVE QUERY AS USER1.QUERY1 (CONFIRM=NO
```

**SET GLOBAL command**

The SET GLOBAL command sets the values of existing global variables or creates new variables and values.
Any new global variables created exist for the entire DataQuant session unless manually deleted.

**Syntax**

```sql
+++ SET GLOBAL ( VariableName = Value, ... ----------
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VariableName</td>
<td>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.</td>
</tr>
<tr>
<td>Value</td>
<td>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 <a href="#">Working with Global Variables</a> for more information.</td>
</tr>
</tbody>
</table>

**Example**

SET GLOBAL (Var1=abc, Var2=def)

### SET INVISIBLE command

The **SET INVISIBLE** command hides specified variables from the Enter Substitution Variable Values dialog.

The Enter Substitution Variable Values dialog will not request any value for the specified variables. If all local variables are set invisible, the Enter Substitution Variable Values 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**

```sql
+++ SET LOCAL ( VariableName1, VariableName2, ... ----------
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VariableName</td>
<td>The name of variable not to be displayed in the Enter Substitution Variable Values dialog.</td>
</tr>
</tbody>
</table>

**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**

```bash
>>>-- SET LOCAL (VariableName = Value, ... ------<<
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VariableName</td>
<td>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 cannot be created or deleted.</td>
</tr>
<tr>
<td>Value</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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 Enter Substitution Variable Values dialog allows you select one of specified values or enter another value.

**Syntax**

```bash
>>>-- SET LOCAL WITH VALUES (VariableName = Value, ... ------<<
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VariableName</td>
<td>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 cannot be created or deleted.</td>
</tr>
<tr>
<td>Value</td>
<td>The value or list of values to give the local variables. The value(s) specified will be available to choose from in the Enter Substitution Variable Values window. The value can be from 1 to 55 characters long. The values of variables whose names begin with DSQ are restricted.</td>
</tr>
</tbody>
</table>

**Example**

```
SET LOCAL WITH VALUES (Var1=abc
```

**SET OPTIONS command**

The **SET OPTIONS** command specifies procedure execution options.

**Syntax**

```bash
>>>-- SET OPTIONS -------------------------------------------------------------------<<
(+ STOPONERROR = YES|NO) ----------------------------------------------------------+
+- SUPPRESSMESSAGES=ALL|INFORM|ERROR +
```
Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOPONERROR</td>
<td>Specifies whether the procedure stops running when an error occurs. Valid values are as follows:</td>
</tr>
<tr>
<td></td>
<td>• YES</td>
</tr>
<tr>
<td></td>
<td>The procedure stops running when an error occurs.</td>
</tr>
<tr>
<td></td>
<td>• NO</td>
</tr>
<tr>
<td></td>
<td>The procedure does not stop running when an error occurs.</td>
</tr>
<tr>
<td></td>
<td>The processing associated with this parameter is similar to the processing associated with DSQQW_PROC_FAIL_ON_ERROR. However, the value</td>
</tr>
<tr>
<td></td>
<td>set for STOPONERROR takes precedent over the values set for DSQQW_PROC_FAIL_ON_ERROR.</td>
</tr>
<tr>
<td></td>
<td>If you do not specify a value for STOPONERROR, the value for DSQQW_PROC_FAIL_ON_ERROR determines whether processing stops or continues</td>
</tr>
<tr>
<td></td>
<td>when an error occurs.</td>
</tr>
<tr>
<td></td>
<td>If you do specify a value for STOPONERROR, the value for DSQQW_PROC_FAIL_ON_ERROR is ignored.</td>
</tr>
<tr>
<td>SUPPRESSMESSAGES</td>
<td>Specifies the types of messages to suppress while the procedure is running. Valid values are as follows:</td>
</tr>
<tr>
<td></td>
<td>• ALL</td>
</tr>
<tr>
<td></td>
<td>Suppresses all message dialogs when the procedure is running.</td>
</tr>
<tr>
<td></td>
<td>• INFORM</td>
</tr>
<tr>
<td></td>
<td>Suppresses information (confirmation) message dialogs when the procedure is running.</td>
</tr>
<tr>
<td></td>
<td>• ERROR</td>
</tr>
<tr>
<td></td>
<td>Suppresses error message dialogs when the procedure is running.</td>
</tr>
</tbody>
</table>

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 FORM and SHOW PROC commands activate the window containing the current query, form, or procedure, respectively. The SHOW REPORT command is a synonym for SHOW FORM command. The SHOW GLOBALS command opens the Global Variables window.

Syntax

```bash
>>-- SHOW PROC --------------------------------<><<
+--GLOBALS ++
+--FORM ----+
+--.MAIN------
+--.BREAK1----
+--.BREAK2----
+--.BREAK3----+
```
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.

The USE REPOSITORY command activate the window containing the current query, form, or procedure, respectively. The SHOW REPORT command is a synonym for SHOW FORM command. The SHOW GLOBALS command opens the Global Variables window.

Syntax

```bash
>>> 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**

```plaintext
USE REPOSITORY Default
USE REPOSITORY SomeRepository (USER=User PASSWORD=password)
```
Chapter 10. Working with global variables

Global variables are variables that stay active while the current session of DataQuant 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 DataQuant application. That means their values are re-initialized each time you open the DataQuant 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 DataQuant 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.

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


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 DataQuant 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.

To edit system global variable values:

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**.
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 DataQuant 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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSQQW_AUTOMATION</td>
<td>1</td>
<td>Indicates whether the application was started as an Automation server.</td>
</tr>
<tr>
<td>DSQQW_CONNECTIONS</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>Name</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DSQQW_DQ</td>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>DSQQW_EXP_DT_FRMT</td>
<td>1</td>
<td>The format to use when exporting data with the EXPORT DATA command in a procedure. Specify the value of: • 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.</td>
</tr>
<tr>
<td>DSQQW_EXP_OUT_MDE</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_FST_SV_DATA</td>
<td>1</td>
<td>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 125.</td>
</tr>
<tr>
<td>DSQQW_HTML_REFTEXT</td>
<td>55</td>
<td>The text that appears in a report when the &amp;REF variable is used. The default value is “Back To”.</td>
</tr>
<tr>
<td>DSQQW_ORIENTATION</td>
<td>0</td>
<td>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.</td>
</tr>
<tr>
<td>Name</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DSQQW_PROC_FAIL_ON_ERROR</td>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>DSQQW_PROC_OUTPUT</td>
<td></td>
<td>Output file name for a procedure.</td>
</tr>
<tr>
<td>DSQQW_PROC_WNDWS</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_QUERY_LANG</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_QUERY_PREP</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_QUERY_PRESERVE_SORT</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_REUSE_OBJC</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>Name</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DSQQW_RPT_COPIES</td>
<td>10</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_FONT</td>
<td>55</td>
<td>Specifies the font face name to use when printing a report with the PRINT REPORT command in a procedure. The default value is &quot;Monospaced&quot;.</td>
</tr>
<tr>
<td>DSQQW_RPT_FONT_BD</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_FONT_CS</td>
<td>3</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_FONT_IT</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_FONT_SZ</td>
<td>2</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_LEN_TYP</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_NUM_CHR</td>
<td>10</td>
<td>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).</td>
</tr>
<tr>
<td>Name</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DSQQW_RPT_NUM_LNS</td>
<td>10</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_ORIENT</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_OUT_TYP</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_TD_TYP</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_TT_TYP</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_USE_PS</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_RPT_WID_TYP</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>Name</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DSQQW_SHOW_QUERY</td>
<td>1</td>
<td>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 diagram view, or three (3) for layout view. The default value is zero (0).</td>
</tr>
<tr>
<td>DSQQW_SQ</td>
<td>1</td>
<td>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 (&quot;).</td>
</tr>
<tr>
<td>DSQQW_SV_DATA_C_S</td>
<td>10</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_SV_DATA_T_M</td>
<td>1</td>
<td>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).</td>
</tr>
<tr>
<td>DSQQW_UEDIT_JAR</td>
<td>55</td>
<td>The name of the JAVA archive file that contains user edit routines.</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSQAO_BATC</td>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>DSQAO_CONNECT_ID</td>
<td>8</td>
<td>The user ID that is used to connect to the current database.</td>
</tr>
<tr>
<td>DSQAO_CURSOR_OPEN</td>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>Name</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DSQAO_DBCS</td>
<td>1</td>
<td>DBCS support status. Value can be one (1) if DBCS support is present or two (2) if DBCS support is not present.</td>
</tr>
<tr>
<td>DSQAO_HOME_WORKSPACE</td>
<td>128</td>
<td>The current repository user's home workspace key, if the workspace exists. Valid values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>rsbi:.workspaces/&lt;user name&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the value if the user connected to a secured repository connection and if the <code>rsbi:.workspaces/&lt;user name&gt;</code> object is viewable by the current user in the repository. <code>&lt;user name&gt;</code> is the login name of the repository user.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>rsbi:.workspaces</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the value if the user connected to a repository connection without security, or the <code>rsbi:.workspaces/&lt;user name&gt;</code> is not viewable by the user, or it does not exist.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>blank</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is value if the previous situations do not exist. For example, if the user is not connected to any repository connection.</td>
</tr>
<tr>
<td><strong>Note:</strong> 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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSQAO_NLF_LANG</td>
<td>1</td>
<td>National language of session. Value is &quot;E&quot; for the English language.</td>
</tr>
<tr>
<td>DSQAO_NUM_FETCHED</td>
<td>0</td>
<td>The number of rows fetched by the current query object.</td>
</tr>
<tr>
<td>DSQAO_OBJ_NAME</td>
<td>18</td>
<td>The name of the current query, form, or procedure object. If there is no current object, the value is blank.</td>
</tr>
<tr>
<td>DSQAO_OBJ_OWNER</td>
<td>8</td>
<td>The owner of the current query, form, or procedure object. If there is no current object, the value is blank.</td>
</tr>
<tr>
<td>DSQAO_REP_USER</td>
<td>8</td>
<td>The user name that is used to connect to the current repository.</td>
</tr>
<tr>
<td>DSQAO_RSBI_RELEASE</td>
<td>2</td>
<td>Numeric release number of the application.</td>
</tr>
<tr>
<td>DSQAO_RSBI_VER_RLS</td>
<td>10</td>
<td>External version and release number for the application.</td>
</tr>
<tr>
<td>DSQAO_QUERY_MODEL</td>
<td>1</td>
<td>Model of the current query object. Value can be one (1) for relational.</td>
</tr>
<tr>
<td>DSQAO_QRY_SUBTYPE</td>
<td>1</td>
<td>Subtype of the current query object. Value can be one (1) for SQL queries or three (3) for queries in a prompted view.</td>
</tr>
</tbody>
</table>
DSQAO_SYSTEM_ID | 1 | Current® operating system. Values can be one of the following:

- 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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
</table>
| DSQEC_CON_ACC_RES  | 1      | For executable SELECT queries that the application submits to DB2 for z/OS, this variable allows you to specify how you want the database to proceed when the data to be selected is locked by an insert, update, or delete operation. When you set this variable, the application specifies the clause associated with the variable value on the concurrent-access-resolution attribute of the PREPARE statement for the SELECT query. Executable SELECT queries can result not only from queries (such as SQL SELECT queries, prompted queries, or QBE P. queries), but also from other operations such as DISPLAY TABLE. Possible values are:
- 0 - No concurrent access resolution options on the PREPARE statement associated with the pending SQL SELECT statement are specified. This is the default value.
- 1 - SKIP LOCKED DATA. This value can be specified for executable SELECT statements directed to DB2 for z/OS Version 9 or later.
- 2 - USE CURRENTLY COMMITTED. This value can be specified for executable SELECT statements directed to DB2 for z/OS Version 10 and later.
- 3 - WAIT FOR OUTCOME. This value can be specified for executable SELECT statements directed to DB2 for z/OS Version 10 and later. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSQEC_EXTND_STG</td>
<td>31</td>
<td>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.</td>
</tr>
<tr>
<td>DSQEC_FORM_LANG</td>
<td>1</td>
<td>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).</td>
</tr>
</tbody>
</table>
| DSQEC_LAST_RUN        | 1      | Specifies the set of commands that cause the LAST_USED column of the Q.OBJECT_DIRECTORY table to be updated. Possible values are:  
* 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    | 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:  
SET GLOBAL (DSQEC_LIST_OWNER=RO%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:  
SET GLOBAL (DSQEC_LIST_OWNER=I_% |
<p>| DSQEC_NLFCMD_LANG     | 1      | 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). |
| DSQEC_RESET_RPT       | 1      | 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>
<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
</table>
| DSQEC_RUN_MQ       | 1      | Specifies whether the RUN QUERY command supports multiple statements in an SQL query. Possible values are:  
                      • 0 - Multiple SQL statements are not supported.  
                         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.  
                      • 1 - Multiple SQL statements are supported.  
                         This is the default value.  
                         A semicolon can be placed at the end of each statement.  
                         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:  
                         --SET STATEMENT DELIMITER="!"  
                         select * from q.staff!  
                         select * from q.org |
| 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:  
                      • 0 - SQL queries directed to DB2 for z/OS, iSeries, 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. The maximum supported query size varies depending on the type of database to which the query is directed:  
                         – Queries directed to DB2 for z/OS can be up to 2 MB in length.  
                         – Queries directed to DB2 for iSeries or DB2 for Linux, UNIX, and Windows can be up to 65 KB in length.  
                         These maximums assume 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:
- 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 11. 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:

<table>
<thead>
<tr>
<th>Edit Code</th>
<th>Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Display character data.</td>
<td>No special formatting.</td>
</tr>
<tr>
<td>CW</td>
<td>Display character data with wrapping based on column width.</td>
<td>No special formatting unless the value cannot 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.</td>
</tr>
<tr>
<td>CT</td>
<td>Display character data with wrapping based on the text in the column.</td>
<td>No special formatting unless the value cannot 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.</td>
</tr>
<tr>
<td>CDx</td>
<td>Display character data with wrapping based on the specified delimiter.</td>
<td>No special formatting unless the value cannot 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 &quot;x&quot; can be any single character, including a blank. The delimiter character does not appear in the report.</td>
</tr>
<tr>
<td>X</td>
<td>Formats data as a series of hexadecimal characters.</td>
<td>No special formatting applied.</td>
</tr>
</tbody>
</table>
### 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:

<table>
<thead>
<tr>
<th>Edit Code</th>
<th>Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDYx</td>
<td>YYYYxMMxDD</td>
<td>Year First</td>
</tr>
<tr>
<td>TDMx</td>
<td>MMxDDxYYYY</td>
<td>Month first</td>
</tr>
<tr>
<td>TDDx</td>
<td>DDxMMxYYYY</td>
<td>Day first</td>
</tr>
<tr>
<td>TDYAx</td>
<td>YYxMMxDD</td>
<td>Two digit year first.</td>
</tr>
<tr>
<td>TDMAx</td>
<td>MMxDDxYY</td>
<td>Month first and two digit year.</td>
</tr>
<tr>
<td>TDDAx</td>
<td>DDxMMxYY</td>
<td>Day first and two digit year.</td>
</tr>
<tr>
<td>TDL</td>
<td>Locally defined</td>
<td>Formats date according to the format specified as the default at the database server requesting data.</td>
</tr>
<tr>
<td>TDT</td>
<td>Locally defined</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Edit Code</th>
<th>Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>XW</td>
<td>Formats data as a series of hexadecimal characters with wrapping based on column width.</td>
<td>Columns are wrapped according to the rules specified for the CW edit code.</td>
</tr>
<tr>
<td>B</td>
<td>Formats data as a series of zeros and ones.</td>
<td>No special formatting applied.</td>
</tr>
<tr>
<td>BW</td>
<td>Formats data as a series of zeros and ones with wrapping based on column width.</td>
<td>Columns are wrapped according to the rules specified for the CW code.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Edit Code</th>
<th>Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>Display graphic data.</td>
<td>No special formatting.</td>
</tr>
<tr>
<td>GW</td>
<td>Display graphic data with wrapping based on column width.</td>
<td>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.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Edit Code</th>
<th>Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Displays numbers in scientific notation.</td>
<td>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.</td>
</tr>
<tr>
<td>EZ</td>
<td>Displays numbers in scientific notation with zero values in the column suppressed.</td>
<td>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.</td>
</tr>
<tr>
<td>Dnn</td>
<td>Displays numbers in decimal notation formatted with a negative sign, thousands separator, and currency symbol.</td>
<td>The value &quot;nn&quot; 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.</td>
</tr>
<tr>
<td>DZnn</td>
<td>Displays numbers in decimal notation formatted with a negative sign, thousands separator, currency symbol and any zero values in the column suppressed.</td>
<td>The value &quot;nn&quot; 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.</td>
</tr>
<tr>
<td>Edit Code</td>
<td>Format</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>DCnn</td>
<td>Displays numbers in decimal notation formatted with a negative sign, thousands separator, and a user-defined currency symbol.</td>
<td>The value &quot;nn&quot; 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.</td>
</tr>
<tr>
<td>DZCnn</td>
<td>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.</td>
<td>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”.</td>
</tr>
<tr>
<td>Inn</td>
<td>Displays numbers in decimal notation formatted with leading zeros displayed and a negative sign.</td>
<td>The value &quot;nn&quot; 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.</td>
</tr>
<tr>
<td>IZnn</td>
<td>Displays numbers in decimal notation formatted with leading zeros displayed, a negative sign and any zero values in the column suppressed.</td>
<td>The value &quot;nn&quot; 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.</td>
</tr>
<tr>
<td>Jnn</td>
<td>Displays numbers in decimal notation formatted with leading zeros displayed.</td>
<td>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.</td>
</tr>
<tr>
<td>JZnn</td>
<td>Displays numbers in decimal notation formatted with leading zeros displayed and any zero values in the column suppressed.</td>
<td>The value &quot;nn&quot; 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.</td>
</tr>
<tr>
<td>Knn</td>
<td>Displays numbers in decimal notation formatted with a negative sign and a thousands separator.</td>
<td>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.</td>
</tr>
<tr>
<td>Edit Code</td>
<td>Format</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>KZnn</td>
<td>Displays numbers in decimal notation formatted with a negative sign, a thousands separator and any zero values in the column suppressed.</td>
<td>The value &quot;nn&quot; 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.</td>
</tr>
<tr>
<td>Lnn</td>
<td>Displays numbers in decimal notation formatted with a negative sign.</td>
<td>The value &quot;nn&quot; 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.</td>
</tr>
<tr>
<td>LZnn</td>
<td>Displays numbers in decimal notation formatted with a negative sign and any zero values in the column suppressed.</td>
<td>The value &quot;nn&quot; 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.</td>
</tr>
<tr>
<td>Pnn</td>
<td>Displays numbers in decimal notation formatted with a negative sign, a thousands separator, and a percent sign.</td>
<td>The value &quot;nn&quot; 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.</td>
</tr>
<tr>
<td>PZnn</td>
<td>Displays numbers in decimal notation formatted with a negative sign, a thousands separator, a percent sign and any zero values in the column suppressed.</td>
<td>The value &quot;nn&quot; 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.</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Edit Code</th>
<th>Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTSx</td>
<td>HHxMMxSS</td>
<td>24 hour clock, including seconds</td>
</tr>
<tr>
<td>TTCx</td>
<td>HHxMMxSS</td>
<td>12 hour clock, including seconds</td>
</tr>
<tr>
<td>TTAx</td>
<td>HHxMM</td>
<td>Abbreviated 24 hour clock (no seconds)</td>
</tr>
<tr>
<td>TTAN</td>
<td>HHMM</td>
<td>Abbreviated 24 hour clock (no seconds) without any delimiter between the time values.</td>
</tr>
<tr>
<td>TTUX</td>
<td>HHxMM PM HHxMM AM</td>
<td>USA format</td>
</tr>
</tbody>
</table>
### 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:

<table>
<thead>
<tr>
<th>Edit Code</th>
<th>Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSI</td>
<td>yyyy–mm–dd–hh.mm.ss.nnnnnnnnn</td>
<td>Where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• yyyy is the four digit year</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• mm is the two digit month</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dd is the two digit day</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• hh is the two digit hour</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• mm is the two digit minute</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ss is the two digit second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• nnnnnnnnnnn is the twelve digit fractional seconds</td>
</tr>
<tr>
<td>Edit Code</td>
<td>Format</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------</td>
<td>-------</td>
</tr>
</tbody>
</table>
| TSZ       | yyyy–mm–dd–hh.mm.ss.nnnnnnnnnn±th:tm | Where:  
- 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  
**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. |

**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:

<table>
<thead>
<tr>
<th>Edit Code</th>
<th>Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSSN</td>
<td>xxx–xx–xxxx</td>
<td>Social security number format.</td>
</tr>
<tr>
<td>VTEL</td>
<td>(xxx) xxx–xxxx</td>
<td>Telephone number format.</td>
</tr>
<tr>
<td>VTEL2</td>
<td>xxx.xxx.xxxx</td>
<td>Telephone number format.</td>
</tr>
<tr>
<td>VZIP</td>
<td>xxxxx–xxxx</td>
<td>Zip code format.</td>
</tr>
</tbody>
</table>

**Implementing custom edit codes for classic reports**

Users can implement custom edit codes that can be used when generating classic reports.

To implement custom edit codes:

1. Create a new Java project and ensure that the Java build path contains the DataQuant 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.getCharacterWidth(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\DataQuant for Workstation.

5. Start DataQuant 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.

```java
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)};
    }
    }
}
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();
}

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();
}

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();
}

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.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.

<table>
<thead>
<tr>
<th>Edit Code</th>
<th>Format</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Displays in character format a column's metadata.</td>
<td>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.</td>
</tr>
</tbody>
</table>
Chapter 12. Customizing the query environment

There are many ways that you can customize your query and reporting environment.

Using the command line

The command line interface allows DataQuant for Workstation users to connect to a repository and run objects by submitting commands using their operating command line rather than the application's GUI interface.

The following steps describe how to connect to a repository and run objects using the operating system command line:

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\IBM DataQuant\DataQuant for Workstation\dataquant.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:
     ```
     ```
   - 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

339
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\IBM DataQuant\DataQuant for Workstation\dataquant.exe" /RConnection:Connection1 /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 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\IBM DataQuant\DataQuant for Workstation\dataquant.exe" /RConnection:Connection1 c:\Procedure1.prc /IServer:DB2AIX /Run /IUserID: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.

*Table 24. Command line parameters*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>Use this parameter if the object that you will run resides in a local or network file. You must specify the full path name.</td>
</tr>
<tr>
<td>/IObject:“ObjectOwner”.“ObjectName”</td>
<td>Use this parameter if the object that you will run resides in a QMF catalog.</td>
</tr>
<tr>
<td>/ROBJECT:pathtooobject</td>
<td>Use this parameter if the object that you will run resides in a repository. You must specify the path to the object. For example: rsb1:/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.</td>
</tr>
<tr>
<td>/RConnection:connectionname</td>
<td>Name of the repository connection that will be used to connect to a repository.</td>
</tr>
<tr>
<td>/IServer:servername</td>
<td>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.</td>
</tr>
<tr>
<td>/Run</td>
<td>Start an opened or retrieved object. Note: If a form object does not have an associated data source, the /Run parameter is ignored.</td>
</tr>
<tr>
<td>/IDisplay</td>
<td>Display the object. If the /Run or /IDisplay parameters are not specified, then the /IDisplay parameter will be used.</td>
</tr>
</tbody>
</table>
### Table 24. Command line parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/RUser:userID</td>
<td>The user ID that will be used to connect to the repository.</td>
</tr>
<tr>
<td>/RPassword:rpassword</td>
<td>The password that will be used with the user ID specified in the /RUser parameter.</td>
</tr>
<tr>
<td>/IUserID:userID</td>
<td>The user ID that will be used to access the data source against which the object will be run.</td>
</tr>
<tr>
<td>/Password:password</td>
<td>The password that will be used with the user ID specified in the /IUserID parameter.</td>
</tr>
</tbody>
</table>
| /Batch                     | 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. |
| "&varianlename=variablevalue" | 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 (&). |
| /CatalogUserID:cataloguserID | The user ID that will be used to access an extended QMF catalog.             |
| /CatalogPassword:catalogpassword | The user password associated with the /CatalogUserID parameter.            |

### Using the command bar

You can run procedure commands directly from the DataQuant for Workstation interface using the command bar.

To run a procedure command from the command bar:

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 DataQuant for Workstation, you must specify the page setup options that will be used when you print the contents of the available window.

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:
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

DataQuant resource limits control your data source access and resource consumption.

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:
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 DataQuant 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 DataQuant 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 DataQuant 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.

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 cancelled.

**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 cancelled.

**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
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 DataQuant 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 DataQuant 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.

<table>
<thead>
<tr>
<th>Confirm database updates value</th>
<th>Procedure parameter value</th>
<th>The confirmation dialog behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected confirm=yes</td>
<td>confirm=no</td>
<td>appears</td>
</tr>
<tr>
<td>Selected confirm=no</td>
<td>confirm=yes</td>
<td>does not appear</td>
</tr>
<tr>
<td>Selected not specified</td>
<td>confirm=yes</td>
<td>appears</td>
</tr>
<tr>
<td>Cleared confirm=yes</td>
<td>confirm=no</td>
<td>appears</td>
</tr>
<tr>
<td>Cleared confirm=no</td>
<td>not specified</td>
<td>does not appear</td>
</tr>
<tr>
<td>Cleared not specified</td>
<td></td>
<td>does not appear</td>
</tr>
</tbody>
</table>

**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_VIEW 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 user's ability to access, retrieve, and save data to and from tables containing large objects (LOBs).

A Large Object (LOB) is a DB2 for z/OS and 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 minutes and hours. After this period, the query result set data will be deleted.

**Cache options can be overridden**

This field specifies whether the cache options can be overridden.
Appendix A. Accessibility

Software accessibility features help those with physical disabilities, such as restricted mobility or limited vision, use their computers.

Note: The DataQuant for Workstation application provides the best accessibility functionality. Users with physical disabilities who must make use of the DataQuant accessibility features should install and run DataQuant for Workstation, as opposed to DataQuant for WebSphere.

Standard keyboard equivalents in DataQuant 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. DataQuant supports the standard keyboard equivalents that include:

- Shortcuts, 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. DataQuant 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.

DataQuant applications:
- Inherit their settings from the operating system
• Do not interfere with keyboard accessibility features built into the operating system

Assistive technology products

DataQuant applications support assistive technology products, such as screen readers and voice synthesizers.

Each DataQuant application interface requires special navigation when using a screen reader with query results. See “Navigation in DataQuant for Workstation” or “Navigation in DataQuant for WebSphere” on page 356 for more information.

Navigation in DataQuant for Workstation

The following table shows how to accomplish various tasks using keystrokes.

<table>
<thead>
<tr>
<th>To:</th>
<th>Process:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate mnemonics in menus</td>
<td>Press the Alt key. Mnemonics in each menu are activated.</td>
</tr>
<tr>
<td>Obtain online help</td>
<td>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.</td>
</tr>
<tr>
<td>Display keyboard shortcuts</td>
<td>Press Ctrl+Shift+L.</td>
</tr>
<tr>
<td>Left to right switch between editor windows</td>
<td>Press Ctrl+F6</td>
</tr>
<tr>
<td>Right to left switch between editor windows</td>
<td>Press Ctrl+Shift+F6</td>
</tr>
<tr>
<td>Move focus through each element</td>
<td>Press TAB to move forward or SHIFT+TAB to move in the opposite direction.</td>
</tr>
<tr>
<td>Emulate clicks on a link</td>
<td>Use the TAB key to jump from link to link, then press the Enter key to open the link.</td>
</tr>
<tr>
<td>Emulate clicks on a button</td>
<td>Use the TAB key to jump from button to button, then press Space key to activate the button.</td>
</tr>
<tr>
<td>Within a window, activate the default action</td>
<td>Press the Enter key.</td>
</tr>
<tr>
<td>Within a window, cancel the action</td>
<td>Press the ESC key.</td>
</tr>
<tr>
<td>Left to right switch between views</td>
<td>Press Ctrl+F7</td>
</tr>
<tr>
<td>Right to left switch between views</td>
<td>Press Ctrl+Shift+F7</td>
</tr>
<tr>
<td>Left to right switch between perspectives</td>
<td>Press Ctrl+F8</td>
</tr>
<tr>
<td>Right to left switch between perspectives</td>
<td>Press Ctrl+Shift+F8</td>
</tr>
<tr>
<td>Left to right switch between SQL, Diagram, Prompted and Results views in the active editor</td>
<td>Press Ctrl+Page Up</td>
</tr>
<tr>
<td>To:</td>
<td>Process:</td>
</tr>
<tr>
<td>-------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Right to left switch between SQL, Diagram, Prompted and Results</td>
<td>Press Ctrl+Page Down</td>
</tr>
<tr>
<td>views in active editor</td>
<td></td>
</tr>
<tr>
<td>Activate command bar</td>
<td>Press Ctrl+Shift+C</td>
</tr>
<tr>
<td>Use screen reader for query results</td>
<td>1. Run the query.</td>
</tr>
<tr>
<td>or</td>
<td>2. Select Alt+R to display the Results menu.</td>
</tr>
<tr>
<td>Use browser to display query results in high contrast mode for the</td>
<td>3. Select Export.</td>
</tr>
<tr>
<td>visually impaired</td>
<td>4. The Export Query Results wizard opens.</td>
</tr>
<tr>
<td></td>
<td>5. Use the TAB key to move through controls. Use arrow keys to move</td>
</tr>
<tr>
<td></td>
<td>through selection items in the control. Select HTML from the Export</td>
</tr>
<tr>
<td></td>
<td>Type field. Specify where to save the file in the File name field.</td>
</tr>
<tr>
<td></td>
<td>7. Open the *.htm file in a browser.</td>
</tr>
<tr>
<td></td>
<td>or</td>
</tr>
<tr>
<td></td>
<td>1. Run the query.</td>
</tr>
<tr>
<td></td>
<td>2. Select Alt+R to display the Results menu.</td>
</tr>
<tr>
<td></td>
<td>3. Select Export.</td>
</tr>
<tr>
<td></td>
<td>4. The Export Query Results wizard opens.</td>
</tr>
<tr>
<td></td>
<td>5. Use the TAB key to move through controls. Use arrow keys to move</td>
</tr>
<tr>
<td></td>
<td>through selection items in the control. Select CSV from the Export</td>
</tr>
<tr>
<td></td>
<td>Type field. Specify where to save the file in the File name field.</td>
</tr>
<tr>
<td></td>
<td>7. Open the *.csv file in supportive software such as Microsoft Excel</td>
</tr>
<tr>
<td></td>
<td>or Lotus® 1-2-3®.</td>
</tr>
<tr>
<td>Add object or condition</td>
<td>1. Tab to put focus on a pane, such as in the Prompted Query dialog.</td>
</tr>
<tr>
<td></td>
<td>2. Tab to put focus on the Add icon.</td>
</tr>
<tr>
<td></td>
<td>3. Press the spacebar to display the Add dialog.</td>
</tr>
<tr>
<td>Select multiple objects</td>
<td>1. Tab to the list box.</td>
</tr>
<tr>
<td></td>
<td>2. Press Shift+Arrow to select additional, consecutive rows.</td>
</tr>
<tr>
<td></td>
<td>3. Press Shift+Enter to add the objects.</td>
</tr>
<tr>
<td>Edit rows in list boxes (only for rows that can be edited)</td>
<td>1. TAB to the list box, such as the Global Variables window.</td>
</tr>
<tr>
<td></td>
<td>2. Press the spacebar or use the up and down arrow keys to select the</td>
</tr>
<tr>
<td></td>
<td>row.</td>
</tr>
<tr>
<td></td>
<td>3. Press F2 to activate edit mode.</td>
</tr>
<tr>
<td></td>
<td>4. Use TAB to move between columns and rows.</td>
</tr>
<tr>
<td></td>
<td>5. Press Enter to accept edits.</td>
</tr>
<tr>
<td></td>
<td>6. Press TAB to exit the list box and move to the next control in the</td>
</tr>
<tr>
<td></td>
<td>dialog.</td>
</tr>
</tbody>
</table>
To: Edit query results

<table>
<thead>
<tr>
<th>Process:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Press <strong>CTRL+F7</strong> or <strong>CTRL+SHIFT+F7</strong> until either the <strong>Workspaces</strong> or <strong>Repository Explorer</strong> view is highlighted.</td>
</tr>
<tr>
<td>2. Use the down or up arrow keys to select the folder that contains a table.</td>
</tr>
<tr>
<td>3. Use the right arrow key to expand folders and the left arrow key to collapse folders.</td>
</tr>
<tr>
<td>4. Use the arrow keys to expose and select the table that you want to edit.</td>
</tr>
<tr>
<td>5. Press the <strong>Context Menu</strong> key (the key to the left of the right-side of the <strong>CTRL</strong> key).</td>
</tr>
<tr>
<td>6. Press the mnemonic key for Open With (the <strong>h</strong> key).</td>
</tr>
<tr>
<td>7. Use the down arrow key to highlight <strong>Table Editor</strong>.</td>
</tr>
<tr>
<td>8. Press the <strong>ENTER</strong> key. The table opens in <strong>Table Editor</strong>.</td>
</tr>
</tbody>
</table>

To: Change font for query results

<table>
<thead>
<tr>
<th>Process:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Run the query.</td>
</tr>
<tr>
<td>2. Press <strong>Alt+A</strong> to select all.</td>
</tr>
<tr>
<td>3. Press <strong>Alt+R</strong> to display the <strong>Results</strong> menu.</td>
</tr>
<tr>
<td>4. Press <strong>F</strong> to display the Font window.</td>
</tr>
</tbody>
</table>

---

**Navigation in DataQuant for WebSphere**

The following table shows how to accomplish various tasks using keystrokes.

<table>
<thead>
<tr>
<th>To: Move focus through each element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process:</td>
</tr>
<tr>
<td>Press <strong>TAB</strong> to move forward or <strong>SHIFT+TAB</strong> to move in the opposite direction.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: Emulate clicks on a link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process:</td>
</tr>
<tr>
<td>Use the <strong>TAB</strong> key to jump from link to link, then press the <strong>Enter</strong> key to open the link.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: Emulate clicks on a button</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process:</td>
</tr>
<tr>
<td>Use the <strong>TAB</strong> key to jump from button to button, then press <strong>Space</strong> key to activate the button.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: Within a dialog, activate the default action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process:</td>
</tr>
<tr>
<td>Press the <strong>Enter</strong> key.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: Within a dialog, cancel the action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process:</td>
</tr>
<tr>
<td>Press the <strong>ESC</strong> key.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: Move focus to the command line</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process:</td>
</tr>
<tr>
<td>Press <strong>ALT+C</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: Move focus to the main menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process:</td>
</tr>
<tr>
<td>Press the left <strong>ALT</strong> key. Navigate through the menu commands using the arrow keys.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: Open context menus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process:</td>
</tr>
<tr>
<td>Press the <strong>context menu</strong> key if you have an extended keyboard. Press the right <strong>Ctrl</strong> key. Navigate through the menu commands using the arrow keys.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: Activate top-level menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process:</td>
</tr>
<tr>
<td>Press <strong>Shift+Ctrl+1</strong> (<strong>Shift+Ctrl+2</strong>, <strong>Shift+Ctrl+3</strong>, and so on)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: Switch between open editors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process:</td>
</tr>
<tr>
<td>Press <strong>Shift+Ctrl+F6</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: Switch between open views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process:</td>
</tr>
<tr>
<td>Press <strong>Shift+Ctrl+F7</strong></td>
</tr>
</tbody>
</table>
Appendix B. Troubleshooting

Use these topics to diagnose and correct problems that you experience with DataQuant.

Problem displaying images in DataQuant for WebSphere using Internet Explorer 8

Symptoms
Visual report and visual dashboard images do not display properly in DataQuant for WebSphere when using Internet Explorer version 8.

Environment
DataQuant 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.

Problem displaying list of objects after upgrading to DB2 UDB for z/OS V8 that supports long names

Symptoms
Prior to DB2 for z/OS V8, object names had to be 8 characters long CHAR(8). If an object name was less than 8 characters long, DB2 for z/OS made up the difference by appending spaces to the end of the name. For example, if an object name was 5 characters long, DB2 for z/OS added 3 spaces to the end of the name. DB2 for z/OS V8 added support for long names. In DB2 for z/OS V8, object names are defined as VARCHAR(128). Object names that were appended with spaces may still exist in your database. To ensure that object names appended with spaces are searchable, you need to run UPDATE statements on the objects. If you do not run the UPDATE statements, object names that were appended will not display in query results.

Environment
DataQuant for WebSphere and DataQuant for Workstation

Resolving the problem
1. Before implementing the fix, back up the following tables:
   - Q.OBJECT_DIRECTORY
   - Q.OBJECT_REMARKS
   - Q.OBJECT_DATA
2. Run the following UPDATE statements:
Creating a silent installation options file

To run a silent installation of DataQuant for Workstation, you must create an options file that will specify each of the features that will be installed.

A silent installation runs on its own without end user intervention. There is no need for end users to monitor the installation and provide input via dialog boxes. Using a silent installation, you as the administrator can control what features are installed on the user’s machine. To set up a silent installation, you must create a silent installation options file and place the file in the same directory as the DataQuant for Workstation installation executable.

To create the silent installation options file:
1. Copy the repositories connections file and the JDBC driver file to the directory where the application executable resides. Copy the QMF Enterprise Edition Version 8, Version 9 or Version 10 license file (if required) to the directory where the application executable resides. Modify the executable name adding the parameter information that will be used by the installation software to create an options file. The following is an example of what you would enter in the command line for a Windows installation:
   ```
   setupwin32.exe -options-record response_file_name
   ```
   where:
   - `-options` specifies that the install software will run with an options file.
   - `-record` specifies that the install software will run in record mode.
   - `response_file_name` specifies the name of the options file.
   This can be any name. You can specify the file name with or without a path prefix. If you specify the file name with a path, the file will be saved at that location. If you specify the file name without a path, the file will be saved in the directory to which you are currently pointing.

   After the command is run, a file named `installer.properties` will be created in the same directory as the install file and contains the responses recorded during the installation.

2. Using a machine that does not currently have DataQuant for Workstation installed, run the modified application executable from the command line.
3. As the installation proceeds, you will select the features that each user that runs this executable will have installed on their machine.
4. Upon completion of the installation, the application is installed in the directory that you specified during the installation.
   The options file is created and saved in the current directory or the directory that you specified in the path.
5. Modify the executable’s name so that each future user that installs will install in silent mode.
   The following is an example of how you would modify the Windows installation executable:
   ```
   setupwin32.exe -silent -options response_file_name
   ```
   where:
- silent: specifies that the install software will run in silent mode.
- options: specifies that the install software will run with an options file.
- response_file_name: specifies the name of the options file.

This is the name that was specified during the record process. If this file is in the same directory as the executable, you need only specify the file name.

If the file was saved in a different directory, you must specify the full path to the file.

If you want the silent installation information to be included in the user's installation, copy the repositories connections file and the JDBC driver file to the directory where the application executable resides.

The install is completely silent; the product directories will just be created with no notification to the user.

Users can now run the executable or you as the administrator can install the product on each user's machine using distribution software.

---

**DataQuant 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 DataQuant for Workstation Version 1 is opened, API instances automatically revert to QMF for Windows, even after Version 1 has been reopened.

**Causes**
When DataQuant for Workstation Version 1 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 1 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 DataQuant for Workstation Version 1.

**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\IBM DataQuant\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 DataQuant for Workstation and DataQuant for WebSphere.

DataQuant for Workstation and DataQuant for WebSphere messages are numbered as follows:

\textit{CPT}prefix\textit{MESS}number

\textit{CPT} prefix

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.

\textit{MESS}number

The specific number of the error message.

For example: ABC0001

All of the error messages of each component are list below 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 DataQuant 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](http://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 \textit{retrieveMessagesFromServerOnGetMessage} option in the Advanced JDBC Settings window of the Edit Repository Connection wizard.

\begin{tabular}{|l|p{1.5in}|p{5in}|}
\hline
\textbf{API1001} & (0) & \\
\hline
\textbf{Explanation}: & This is a container for other messages. & \\
\textbf{User response}: & No action is required. & \\
\hline
\textbf{API1002} & The API has not been initialized. & \\
\hline
\textbf{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. & \\
\textbf{User response}: & Check the QMFwin API logs to determine the error that caused the API instance to fail, or establish a repository connection. & \\
\hline
\textbf{API1003} & No data found. & \\
\hline
\textbf{Explanation}: & There was an attempt to obtain a row that does not exist in a result set. & \\
\textbf{User response}: & No action required. & \\
\hline
\textbf{API1006} & The specified source (0) is not valid. & \\
\hline
\textbf{Explanation}: & An invalid value was specified for the SourceType parameter of the initializeProc() or initializeQuery() function. & \\
\textbf{User response}: & Specify a valid value for the SourceType parameter of any initializeProc() or initializeQuery() function. & \\
\hline
\textbf{API1007} & The specified source name is not valid. & \\
\hline
\textbf{Explanation}: & An invalid value was specified for the Source, CollectionName, PackageName, or ConsistencyToken parameter of an initializeProc(), initializeQuery(), or initializeStaticQuery() function. & \\
\textbf{User response}: & Specify a valid value for the Source, CollectionName, PackageName, or ConsistencyToken parameter of any initializeProc(), initializeQuery(), or initializeStaticQuery() function. & \\
\hline
\textbf{API1008} & The specified database server (0) is not valid. & \\
\hline
\end{tabular}
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.

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.

API1037  The specified procedure ID ({}{0}) is not valid.
Explanation:  An invalid value was specified for the ProcID parameter or 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.

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 • CLC00003

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 DataQuant
Explanation: The specified feature is not supported by DataQuant
User response: No action is required.

BIC00007 Object is not supported by DataQuant
Explanation: The specified object is not supported by DataQuant
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.
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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

Cannot multiply two measured values.

Explanation: Measured values cannot be multiplied.

User response: Specify values that are not measured to be multiplied.

Cannot create list from elements of "{0}" type.

Explanation: A list cannot be created using elements of the indicated type.

User response: Specify elements of a different type to be listed.

List cannot contain elements of different types.

Explanation: The specified list cannot contain elements of different types.

User response: Specify elements of the same type to be listed.

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.
Argument(s) with measurement units are not acceptable for "{0}" function.

Explanation: The indicated function cannot contain arguments with measurement units.

User response: Remove all measurement units from the arguments of the function.

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.

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.

Calculator exception cause: {0}

Explanation: A generic error occurred.

User response: Check the syntax of the expression for errors and try again.

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.

Function "{0}" caused error: {1}

Explanation: The indicated function caused the indicated error.

User response: Correct the indicated error and rerun the function.

Property "{0}" contains error: {1}

Explanation: The indicated property caused the indicated error.

User response: Correct the indicated error and rerun the function.

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.

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.

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.

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.

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.

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.

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.

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
<table>
<thead>
<tr>
<th>DAL01007</th>
<th>Character encoding [0] is not supported.</th>
<th>DAL01016</th>
<th>Cannot parse data (base64 encoding rules violation)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated character encoding is not supported.</td>
<td><strong>Explanation:</strong></td>
<td>The XML file is corrupted.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Use a character set that is supported by your current JVM.</td>
<td><strong>User response:</strong></td>
<td>Fix any errors in the XML file and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01008</th>
<th>An error occurred while accessing the database.\n(0)</th>
<th>DAL01017</th>
<th>Duplicate query with name &quot;[0]&quot; is detected in package &quot;[1]&quot;.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>An error occurred while accessing the indicated database.</td>
<td><strong>Explanation:</strong></td>
<td>An internal error occurred; the indicated package contains several queries with the same specified name.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help</td>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help</td>
</tr>
<tr>
<td>&gt; Collect Support Data and forward it to customer support.</td>
<td>&gt; Collect Support Data and forward it to customer support.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01009</th>
<th>An internal error occurred. [0]</th>
<th>DAL01018</th>
<th>The source data is not compatible with the target table.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated internal error occurred.</td>
<td><strong>Explanation:</strong></td>
<td>The attempt to save data in the database table failed because of the following:</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help</td>
<td>• Number of columns does not match.</td>
<td>• Columns have incompatible types.</td>
</tr>
<tr>
<td>&gt; Collect Support Data and forward it to customer support.</td>
<td>&gt; Collect Support Data and forward it to customer support.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01011</th>
<th>Class [0] is not suitable for server description</th>
<th>DAL01019</th>
<th>Operation is not supported.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>An internal error occurred.</td>
<td><strong>Explanation:</strong></td>
<td>An internal error occurred; the requested operation can not be performed due to internal limitations.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help</td>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help</td>
</tr>
<tr>
<td>&gt; Collect Support Data and forward it to customer support.</td>
<td>&gt; Collect Support Data and forward it to customer support.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01012</th>
<th>Requested operation is not available in offline mode</th>
<th>DAL01020</th>
<th>Invalid SQL type definition: &quot;[0]&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified operation is not available in offline mode.</td>
<td><strong>Explanation:</strong></td>
<td>An internal error occurred; the indicated SQL type definition is invalid (it can be mapped to more than one known data types).</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify a different operation to apply to the active object, or switch to online mode and try again.</td>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help</td>
</tr>
<tr>
<td>&gt; Collect Support Data and forward it to customer support.</td>
<td>&gt; Collect Support Data and forward it to customer support.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01013</th>
<th>Cannot create connection to server &quot;[0]&quot; due to the following: [1]</th>
<th>DAL01021</th>
<th>This product does not contain SQLj support module</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>A connection to the indicated server could not be established due to the indicated error.</td>
<td><strong>Explanation:</strong></td>
<td>DataQuant does not support or can not initialize the SQLj support module.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Resolve the indicated error and retry to connect to the server.</td>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help</td>
</tr>
<tr>
<td>&gt; Collect Support Data and forward it to customer support.</td>
<td>&gt; Collect Support Data and forward it to customer support.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DAL01014</th>
<th>Invalid set of connection parameters used.</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>An internal error occurred.</td>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help</td>
<td>&gt; Collect Support Data and forward it to customer support.</td>
<td></td>
</tr>
</tbody>
</table>
### 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 saving 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 DataQuant. 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 cannot 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 DataQuant.
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.

DAL01045  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.

DAL01046  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.

DAL01047  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.

DAL01048  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.
<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Explanation</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDM0002</td>
<td>Object ( {0} ) have no fields with codepoint ( {1} )</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0003</td>
<td>Socket open exception. ( {0} )</td>
<td>Unable to establish TCP/IP connection.</td>
<td>Check the host and port details in the JDBC URL; ping the server to check that the network connection works.</td>
</tr>
<tr>
<td>DDM0004</td>
<td>Socket write exception. ( {0} )</td>
<td>There was a network connection failure or an internal error in the driver code was detected.</td>
<td>Check your network connection. If the connection works fine, send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0005</td>
<td>Data format exception detected.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0006</td>
<td>Invalid JDBC URL ( {0} )</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0007</td>
<td>DSS header format error detected.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0008</td>
<td>Unsupported CCSID ( {0} )</td>
<td>An internal error was detected in the driver code.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0009</td>
<td>Cipher error ( {0} )</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0010</td>
<td>Object ( {0} ) does not contain required field ( {1} )</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0011</td>
<td>Object ( {0} ) does not support value ( {1} )</td>
<td>An internal error was detected in the driver code.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0012</td>
<td>Syntax error detected: ( {0} )</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0013</td>
<td>Syntax error detected: ( {0} )</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0014</td>
<td>Object ( {0} ) can hold no more repeated fields ( {1} )</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0015</td>
<td>Illegal port number value: ( {0} )</td>
<td>The JDBC URL contains an invalid value in the Port field.</td>
<td>Specify a correct port number in the JDBC URL.</td>
</tr>
<tr>
<td>DDM0016</td>
<td>( {0} ) is invalid length for the field ( {1} ) of object ( {2} )</td>
<td>An internal error was detected in the driver code.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>DDM0017</td>
<td>Unknown FD:OCA descriptor type id ( {0} )</td>
<td>An internal error was detected in the driver code.</td>
<td>Send an error report to the developers.</td>
</tr>
</tbody>
</table>
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.Types 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 number of connections parameter in the Repository Storage dialog or set the value to zero to remove this restriction.

DSCERRCD0001  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.

DSCERRCD0002  FD:OCA triplet sequence error.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

DSCERRCD0003  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.

DSCERRCD0004  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.

DSCERRCD0005  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.

DSCERRCD0006  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.

DSCERRCD0007  Parameter value is not acceptable.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

DSCERRCD0017  Meta-Data Descriptor (MDD) 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.

DSCERRCD0018 MDD class is not recognized as a valid SQL class.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

DSCERRCD0019 MDD type is not recognized as a valid SQL type.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

DSCERRCD0033 Representation is incompatible 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.

DSCERRCD0034 CCSID is not supported.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

DSCERRCD0050 Group Data Array (GDA) references 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.

DSCERRCD0051 GDA length override exceeds limits.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

DSCERRCD0052 GDA precision exceeds limits.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

DSCERRCD0053 GDA scale greater than precision or scale negative.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

DSCERRCD0054 GDA length override missing or incompatible with data type.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

DSCERRCD0065 RLO references a LID which is not an RLO or GDA.
Explanation: An internal error in the driver code was detected.
User response: Send an error report to the developers.

DSCERRCD0066 RLO fails to reference a required 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.

Appendix C. Messages 375
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.
<table>
<thead>
<tr>
<th>Code</th>
<th>Message Description</th>
<th>Explanation</th>
<th>User Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM0024</td>
<td>RDB Not Accessed.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0025</td>
<td>Data Descriptor Mismatch.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0026</td>
<td>Object Not Supported.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0027</td>
<td>RDB Package Binding Not Active.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0028</td>
<td>RDB Update Reply Message.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0029</td>
<td>SQL Error Condition.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0030</td>
<td>End of Data.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0031</td>
<td>End of Query.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0032</td>
<td>Commitment Request.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0033</td>
<td>End Unit of Work Condition.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0034</td>
<td>Invalid Descriptor.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0035</td>
<td>Open Query Complete.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0036</td>
<td>Open Query Failure.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0037</td>
<td>Query Previously Opened.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM0038</td>
<td>Command Violation.</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>ERM1200</td>
<td>Generic DRDA error has occurred. {0}</td>
<td>The indicated generic DRDA error has occurred.</td>
<td>Resolve the indicated generic DRDA error and try again.</td>
</tr>
<tr>
<td>ERM1201</td>
<td>Generic DRDA error has occurred. {0}, {1}</td>
<td>The two indicated generic DRDA error have occurred.</td>
<td>Resolve the two indicated errors and try again.</td>
</tr>
</tbody>
</table>
ERM1202 • EXPT1005

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 indicated 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.

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 DataQuant 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 DataQuant 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 DataQuant 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 DataQuant 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 DataQuant 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 DataQuant 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 DataQuant 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 then 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 DataQuant 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.
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Explanation</th>
<th>User Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRM1027</td>
<td>There is no column [0].</td>
<td>The expression contains a variable that refers to a column that does not exist.</td>
<td>Specify a valid column number in the expression variable.</td>
</tr>
<tr>
<td>FRM1028</td>
<td>The suffix '_B' is invalid in this context.</td>
<td>The suffix '_B' can not be used in predefined variables, and as such is invalid for the current variable.</td>
<td>Remove the suffix '_B' from all predefined variables.</td>
</tr>
<tr>
<td>FRM1029</td>
<td>Unable to open linked data source.</td>
<td>The form attempted to open a linked data source and an error occurred during this process.</td>
<td>Specify a valid data source object for the form.</td>
</tr>
<tr>
<td>FRM1030</td>
<td>An error occurred while registering user edit code.</td>
<td>The FormCustomEditCode returned an error.</td>
<td>Review the FormCustomEditCode implementation documentation and check for errors.</td>
</tr>
<tr>
<td>FRM1031</td>
<td>The user-defined edit routine class could not be found. [0]</td>
<td>The user implementation class FormCustomEditCode can not be loaded by the Classloader.</td>
<td>Check the className.jar CLASSPATH.</td>
</tr>
<tr>
<td>FRM1032</td>
<td>An error occurred while initializing user-defined edit routine class. [0]</td>
<td>An error occurred.</td>
<td>See additional information in the wrapped exception.</td>
</tr>
<tr>
<td>GERR00003</td>
<td>Generic driver error has occurred. [0], [1], [2], [3]</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>GERR00004</td>
<td>Generic driver error has occurred. [0], [1], [2], [3], [4]</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>GERR00005</td>
<td>Generic driver error has occurred. [0], [1], [2], [3], [4], [5]</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>GERR00006</td>
<td>Generic driver error has occurred. [0], [1], [2], [3], [4], [5], [6]</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>GERR00007</td>
<td>Generic driver error has occurred. [0], [1], [2], [3], [4], [5], [6], [7]</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>GERR00008</td>
<td>Generic driver error has occurred. [0], [1], [2], [3], [4], [5], [6], [7], [8]</td>
<td>An internal error in the driver code was detected.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>IXF01002</td>
<td>The LOB column index is invalid.</td>
<td>The file contains an invalid index of a LOB column.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>IXF01003</td>
<td>The LOB column length is invalid.</td>
<td>The IXF file contains a LOB column that has an invalid length.</td>
<td>Send an error report to the developers.</td>
</tr>
</tbody>
</table>
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 DataQuant.

User response: Create an IXF file using another CCSID to import to DataQuant.

**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 DataQuant.

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: 
User response: 

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 (.jdfs).
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 (.jdfs).
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.
<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Explanation</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLQ1006</td>
<td>Only one reference to dimension hierarchy (0) can appear in the slicer.</td>
<td>The MDX DB2 DWE supports only one dimension per slicer.</td>
<td>Select only one dimension.</td>
</tr>
<tr>
<td>PE00001</td>
<td>Encountered &quot;(0)&quot; at column (1). \r\nWas expecting one of: (2).</td>
<td>An invalid character was encountered at the indicated column.</td>
<td>Replace the invalid character with one of the indicated characters.</td>
</tr>
<tr>
<td>PE00002</td>
<td>Expression needs a calculator to get a value.</td>
<td>The expression does not return a value.</td>
<td>No action is required.</td>
</tr>
<tr>
<td>PE00004</td>
<td>Parse exception occurred.</td>
<td>An error occurred while parsing the expression.</td>
<td>Check the expression syntax for errors and try again.</td>
</tr>
<tr>
<td>PRC1000</td>
<td>An unexpected error occurred while parsing the command.</td>
<td>An internal error occurred.</td>
<td>Collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>PRC1001</td>
<td>An error occurred while parsing the command: unexpected token &quot;(0)&quot;.</td>
<td>An unexpected token was encountered while parsing the procedure.</td>
<td>Check the procedure's text for syntax errors (using the documentation or Content Assist via Ctrl+Space)</td>
</tr>
<tr>
<td>PRC1002</td>
<td>An error occurred while parsing the command: unexpected character &quot;(0)&quot;.</td>
<td>An unexpected character was encountered while parsing the procedure.</td>
<td>Check the procedure's text for syntax errors (using the documentation or Content Assist via Ctrl+Space)</td>
</tr>
<tr>
<td>PRC1003</td>
<td>The parameter &quot;(0)&quot; is invalid in this context.</td>
<td>An unsupported parameter token was encountered while parsing the procedure.</td>
<td>Check the procedure's text for syntax errors (using the documentation or Content Assist via Ctrl+Space)</td>
</tr>
<tr>
<td>PRC1004</td>
<td>The parameter &quot;(0)&quot; cannot be specified twice.</td>
<td>The indicated parameter was specified twice in the same procedure command.</td>
<td>Only specify one instance of the indicated parameter.</td>
</tr>
<tr>
<td>PRC1005</td>
<td>&quot;(1)&quot; is not a valid value for parameter &quot;(0)&quot;.</td>
<td>The indicated value is not valid for the indicated parameter.</td>
<td>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.</td>
</tr>
<tr>
<td>PRC1006</td>
<td>You cannot set variable &quot;(0)&quot;.</td>
<td>The indicated variable can not be set manually.</td>
<td>Specify a valid variable and try again.</td>
</tr>
<tr>
<td>PRC1008</td>
<td>An error occurred while parsing the command: unexpected end of line.</td>
<td>An unexpected end of line was encountered while parsing the procedure.</td>
<td>Check the procedure's text for syntax errors (using the documentation or Content Assist via Ctrl+Space)</td>
</tr>
<tr>
<td>PRC1009</td>
<td>An error occurred while parsing the command: unrecognized command &quot;(0)&quot;.</td>
<td>The indicated command is not valid.</td>
<td>Check the procedure text for errors or specify a different command.</td>
</tr>
</tbody>
</table>
### PRC1010 — Parsing error at lines 0-1 (command #2).

**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].

**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 the 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.

Appendix C. Messages 389
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRC2035</td>
<td>The printer name &quot;[0]&quot; is invalid.</td>
</tr>
<tr>
<td><strong>Explanation:</strong> The indicated printer name is invalid.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> Check the spelling of the printer name, or specify a different printer to use for the PRINT command.</td>
<td></td>
</tr>
<tr>
<td>PRC2036</td>
<td>You are not authorized to use the table editor at this data source.</td>
</tr>
<tr>
<td><strong>Explanation:</strong> The Enable Table Editing resource limit for the data source is turned off; you cannot open Table Editor of the data source.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> Ask your administrator to enable this resource limit.</td>
<td></td>
</tr>
<tr>
<td>PRC2037</td>
<td>You are not authorized to edit tables at this data source.</td>
</tr>
<tr>
<td><strong>Explanation:</strong> The Enable Table Editing resource limit for the data source is turned off; you cannot edit values in a query's grid cells.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> Ask your administrator to enable this resource limit.</td>
<td></td>
</tr>
<tr>
<td>PROC1000</td>
<td>REXX procedure references cannot be determined.</td>
</tr>
<tr>
<td><strong>Explanation:</strong> References for the procedure cannot be built for REXX procedures.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> You can view references at the Repository or Workspaces views for generic procedures only.</td>
<td></td>
</tr>
<tr>
<td>PROC1001</td>
<td>Some substitution variables are not defined.</td>
</tr>
<tr>
<td><strong>Explanation:</strong> Some of the substitution variables of the procedure are not defined.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> 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 DataQuant will ask for all of the necessary variables values.</td>
<td></td>
</tr>
<tr>
<td>QEL00001</td>
<td>You are not allowed to run queries returning LOB data.</td>
</tr>
<tr>
<td><strong>Explanation:</strong> You are not allowed to run queries that contain large object data.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> 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.</td>
<td></td>
</tr>
<tr>
<td>QEL00002</td>
<td>The Maximum Rows to Fetch limit has been exceeded. [0] rows were retrieved so far.</td>
</tr>
<tr>
<td><strong>Explanation:</strong> The Maximum Rows to Fetch limit has been reached. The indicated number of rows have been retrieved so far.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> 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.</td>
<td></td>
</tr>
<tr>
<td>QEL00003</td>
<td>The Maximum Bytes to Fetch limit has been exceeded. [0] bytes were retrieved so far.</td>
</tr>
<tr>
<td><strong>Explanation:</strong> The Maximum Bytes to Fetch limit has been reached. The indicated number of bytes have been retrieved so far.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> 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.</td>
<td></td>
</tr>
<tr>
<td>QEL00004</td>
<td>The length of a LOB column exceeds the maximum allowable length.</td>
</tr>
<tr>
<td><strong>Explanation:</strong> The length of a LOB column in the query results exceeds the maximum allowable length.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> 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.</td>
<td></td>
</tr>
<tr>
<td>QEL00005</td>
<td>You are not allowed to use the &quot;[0]&quot; SQL verb.</td>
</tr>
<tr>
<td><strong>Explanation:</strong> You are not allowed to use the indicated SQL verb.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> 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.</td>
<td></td>
</tr>
<tr>
<td>QEL00006</td>
<td>You are not allowed to save data to database.</td>
</tr>
<tr>
<td><strong>Explanation:</strong> You are not allowed to save data to the database.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> 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.</td>
<td></td>
</tr>
</tbody>
</table>

---

390  Getting Started with DataQuant
<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Explanation</th>
<th>User Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>QEN00001</td>
<td>Query was cancelled.</td>
<td>The active query was cancelled.</td>
<td>Rerun the query.</td>
</tr>
<tr>
<td>QEN00003</td>
<td>Unexpected token: {0}</td>
<td>An unexpected token was encountered while parsing the query that calls the stored procedure.</td>
<td>Check the query's syntax for errors.</td>
</tr>
<tr>
<td>QEN00004</td>
<td>Unexpected token: {0} is expected but {1} was found.</td>
<td>An unexpected closed bracket was encountered while parsing the query that calls the stored procedure.</td>
<td>Check the query's syntax for errors.</td>
</tr>
<tr>
<td>QEN00005</td>
<td>Unexpected end of statement.</td>
<td>An unexpected end of statement was encountered while parsing the query that calls the stored procedure.</td>
<td>Check the query's syntax for errors.</td>
</tr>
<tr>
<td>QEN00006</td>
<td>Expression {0} is not a stored procedure.</td>
<td>An internal error occurred.</td>
<td>Collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>QEN00007</td>
<td>End of statement is expected but {0} was found</td>
<td>Tokens were encountered after the end of statement while parsing the query that calls the stored procedure.</td>
<td>Check the query's syntax for errors.</td>
</tr>
<tr>
<td>QEN00008</td>
<td>Unsupported parameter data type: {0}</td>
<td>An unsupported parameter was encountered while parsing the query that calls the stored procedure.</td>
<td>Check the query's syntax for errors.</td>
</tr>
<tr>
<td>QEN00009</td>
<td>{0} is invalid value for this type: {1}</td>
<td>When prompted for a value for a stored procedure parameter, the user entered a value that is not valid for that parameter's type.</td>
<td>Specify a valid value for the parameter.</td>
</tr>
<tr>
<td>QMF1004</td>
<td>You are not authorized to print objects from &quot;{0}&quot; data source.</td>
<td>The indicated data source name has a restriction on printing for the current user.</td>
<td>Contact your administrator to check your specific permissions for printing.</td>
</tr>
<tr>
<td>QMF1005</td>
<td>QMF Catalog table structure is obsolete.\nCatalog tables must be updated.</td>
<td>The indicated an error occurred because the QMF Catalog has an obsolete structure.</td>
<td>Contact your administrator to upgrade the QMF Catalog.</td>
</tr>
<tr>
<td>QMF1006</td>
<td>You are not authorized to save data to files from &quot;{0}&quot; data source.</td>
<td>The indicated data source name has a restriction on saving data for the current user.</td>
<td>Contact your administrator to check your specific permissions for saving data.</td>
</tr>
<tr>
<td>QRL00001</td>
<td>You are not authorized to access this data source in user interface mode.</td>
<td>You are not authorized to access this data source in user interface mode.</td>
<td>Check your Options limits by selecting Resource Limits from the View menu to see your specific allowances or ask your administrator for more information.</td>
</tr>
<tr>
<td>QRL00002</td>
<td>You are not authorized to access this data source in automation mode.</td>
<td>You are not authorized to access this data source in automation mode.</td>
<td>Check your Options limits by selecting Resource Limits from the View menu to see your specific allowances or ask your administrator for more information.</td>
</tr>
<tr>
<td>QRL00003</td>
<td>You are not authorized to run unsaved queries.</td>
<td>You are not authorized to run unsaved queries.</td>
<td>Check your Options limits by selecting Resource Limits from the View menu to see your specific allowances or ask your administrator for more information.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>Explanation</td>
<td>User Response</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>QRL00004</td>
<td>You are not authorized to save data from this data source to a file.</td>
<td>You are not authorized to save data from this data source to a file.</td>
<td>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.</td>
</tr>
<tr>
<td>QRL00005</td>
<td>You are not authorized to save data from this data source to the database.</td>
<td>You are not authorized to save data from this data source to the database.</td>
<td>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.</td>
</tr>
<tr>
<td>QRL00006</td>
<td>You are not authorized to retrieve LOB data.</td>
<td>You are not authorized to retrieve LOB data.</td>
<td>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.</td>
</tr>
<tr>
<td>QRL00008</td>
<td>No resource group for user {0} was found.</td>
<td>No resource group for the indicated user was found.</td>
<td>Specify a valid resource group for the user.</td>
</tr>
<tr>
<td>QRL00009</td>
<td>The operation cannot be performed because the processing of LOB data is disabled.</td>
<td>The operation cannot be performed because the processing of LOB data is disabled.</td>
<td>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.</td>
</tr>
<tr>
<td>QRY1001</td>
<td>Query was cancelled.</td>
<td>The query was cancelled.</td>
<td>Rerun the query.</td>
</tr>
<tr>
<td>QRY1002</td>
<td>An invalid header record was encountered.</td>
<td>The query was loaded with errors.</td>
<td>Try to reload the query; otherwise the query is corrupted.</td>
</tr>
<tr>
<td>QRY1003</td>
<td>No data source was specified for the query.</td>
<td>No data source was specified for the query.</td>
<td>Specify a data source for the query.</td>
</tr>
<tr>
<td>QRY1004</td>
<td>An unexpected &quot;R&quot; record was encountered: {0}.</td>
<td>An error was encountered while loading a type R record from the prompted query file during storage of info in the prompted query structure.</td>
<td>Collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>QRY1005</td>
<td>An unexpected &quot;T&quot; record was encountered: {0}.</td>
<td>An error was encountered while loading a type T record from the prompted query file during storage of info in the prompted query structure.</td>
<td>Collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>QRY1006</td>
<td>An invalid &quot;T&quot; record was encountered: {0}.</td>
<td>An error was encountered while loading a type T record from the prompted query file during storage of info in the prompted query structure.</td>
<td>Collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>QRY1007</td>
<td>A &quot;T&quot; record with invalid column information was encountered: {0}.</td>
<td>An error was encountered while loading a type T record from the prompted query file during storage of info in the prompted query structure.</td>
<td>Collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
</tbody>
</table>
> 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 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.

**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.
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.

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.

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.

The expression contains a syntax error:

Explanation: The indicated expression contains a syntax error.

User response: Correct the syntax error and rerun the query.

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.

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.

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.

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.

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.

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.

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.

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.

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 table name.
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.
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QRY1045</td>
<td>Only simple row conditions are supported.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>Only simple row conditions are supported.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Ensure that the specified query only contains simple row conditions.</td>
</tr>
<tr>
<td>QRY1046</td>
<td>Expression cannot be empty.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>An error occurred while constructing the query.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Check the query's syntax for errors and try again.</td>
</tr>
<tr>
<td>QRY1047</td>
<td>Invalid type for saving the query.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The type that was set for the query is invalid.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify a valid type for the query or collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>QRY1048</td>
<td>You cannot run an empty query.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified query did not contain any SQL code.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Populate the query with at least a SELECT statement and rerun.</td>
</tr>
<tr>
<td>QRY1049</td>
<td>Some substitution variables are not defined.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>Some of the substitution variables in the query were not given values when the query was run.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Ensure that all of the substitution variables in the query are defined.</td>
</tr>
<tr>
<td>QRY1050</td>
<td>You cannot view a query that uses row values in the predicate.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>An error occurred while parsing the SQL text to construct the prompted query.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Check the SQL text for errors and try again.</td>
</tr>
<tr>
<td>QRY1051</td>
<td>'XMLExists' predicates are not supported.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>An error occurred while parsing the SQL text to construct the prompted query.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Check the SQL text for errors and try again.</td>
</tr>
<tr>
<td>QRY1052</td>
<td>An invalid value: [0].</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>An error occurred while parsing the SQL text to construct the prompted query.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Check the SQL text for errors and try again.</td>
</tr>
<tr>
<td>QRY1053</td>
<td>Only SELECT statements are supported in compound queries.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>Only SELECT statements are supported in compound queries.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Ensure that only SELECT statements are called in the specified compound query.</td>
</tr>
<tr>
<td>QRY1054</td>
<td>Query content is not compatible with Prompted Query format.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>The query can not be saved in the prompted format.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Save the query in another format.</td>
</tr>
<tr>
<td>QRY1055</td>
<td>Host variable predicates are not supported.</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>An error occurred while parsing the SQL text to construct the prompted query.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Check the SQL text for errors and try again.</td>
</tr>
<tr>
<td>RCO0000</td>
<td>Internal error</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>An unexpected error occurred while processing the *.rco file.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>RCO0001</td>
<td>Type mismatch</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>An unexpected error occurred while processing the *.rco file.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>RCO0002</td>
<td>Function [0] is not implemented yet</td>
</tr>
<tr>
<td><strong>Explanation:</strong></td>
<td>An unexpected error occurred while processing the *.rco file.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>Code</td>
<td>Message</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>RCO0003</td>
<td>Undefined variable {0}</td>
</tr>
<tr>
<td>RCO0004</td>
<td>Missing variable name</td>
</tr>
<tr>
<td>RCO0005</td>
<td>Syntax error in &quot;{0}&quot;</td>
</tr>
<tr>
<td>RCO0006</td>
<td>Unknown function {0}</td>
</tr>
<tr>
<td>RCO0007</td>
<td>Unknown object type {0}</td>
</tr>
<tr>
<td>RCO0008</td>
<td>Wrong parameters in &quot;{0}&quot;</td>
</tr>
<tr>
<td>RCO0009</td>
<td>Database object {0} not found</td>
</tr>
<tr>
<td>RCO0100</td>
<td>Database error {0}</td>
</tr>
<tr>
<td>RCO0101</td>
<td>Unknown database server type {0}</td>
</tr>
<tr>
<td>REP01000</td>
<td>{0}</td>
</tr>
<tr>
<td>REP01001</td>
<td>An input/output error occurred: {0}</td>
</tr>
<tr>
<td>REP01002</td>
<td>Character encoding is not supported: {0}</td>
</tr>
<tr>
<td>REP01003</td>
<td>Internal error has occurred: {0}</td>
</tr>
<tr>
<td>REP01004</td>
<td>The following error has occurred while accessing database: {0}</td>
</tr>
<tr>
<td>REP01005</td>
<td>Operation is not supported.</td>
</tr>
</tbody>
</table>

Appendix C. Messages 397
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.
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.
Explanation: An internal error occurred in the repository.
User response: Collect support data by selecting Help > Collect Support Data and forward it to customer support.

REP01018
User is not logged on repository.
Explanation: You cannot perform the specified operation, because you are not connected to the repository.
User response: Connect to the repository and try again.
REP01019  Not implemented.
Explanation: The indicated error occurred while attempting to access a non-implemented functionality.
User response: Ensure that you are using the latest version of the product.

REP01020  Unexpected error occurred.
Explanation: An internal error occurred in the repository.
User response: Collect support data by selecting Help > Collect Support Data and forward it to customer support.

REP01021  Initialization error occurred "{0}".
Explanation: An error occurred while the repository was being initialized.
User response: Check the repository initialization parameters and try again.

REP01022  Invalid initialization parameters were specified "{0}".
Explanation: An error occurred while the repository was being initialized.
User response: Check the repository initialization parameters and try again.

REP01027  A repository instance hasn't been initialized.
Explanation: An error occurred while trying to access a non-initialized repository.
User response: Initialize the repository and try again.

REP01028  Repository "{0}" could not be found.
Explanation: The indicated repository could not be found.
User response: Check the repository name for errors or specify a different repository and try again.

REP01029  Object type "{0}" cannot be added to this folder.
Explanation: Objects of the indicated type can not be added to the specified folder.
User response: Specify a different folder in which to add the indicated object.

REP01030  More authentication information for "{0}" required.
Explanation: Not enough authentication information was provided for the indicated object.
User response: Provide more authentication information to the indicated object and try again.

REP01031  Repository storage table structure is obsolete. Contact repository storage administrator.
Explanation: An error occurred while trying to access a repository that has an obsolete structure.
User response: Contact your administrator to upgrade the repository database structure.

REP01032  CCSID information could not be found for current QMF catalog.
Explanation: An error occurred while trying to read CCSID information from the QMF Catalog.
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.
<table>
<thead>
<tr>
<th>REP01038</th>
<th>Undefined or wrong repository identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>An error occurred while trying to initialize a repository with deprecated initialization parameters.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Change the repository connection properties through the Edit Repository Connection dialog and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REP01039</th>
<th>There are no catalogs on the server</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>No QMF catalogs exist on the specified server.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify a different server and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REP01042</th>
<th>You are not authorized to use objects with Owner = {0}.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>Your user ID is not authorized to use objects with the indicated owner.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify an object that has a different owner or contact your administrator for more information.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REP01043</th>
<th>This data source cannot be accessed by DataQuant</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The application can not access this data source.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Check your user license and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REP01044</th>
<th>Access to data source &quot;{0}&quot; blocked due to QMF catalog restriction violation.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>Access to the indicated data source is blocked due to a restriction imposed by the QMF catalog protection limits.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify a different data source to access or contact your administrator for more information.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REP01045</th>
<th>Associated QMF catalog data source could not be found: {0}</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated QMF catalog data source could not be found.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Check the data source name for errors or specify a different data source and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REP01046</th>
<th>Cannot find connection: {0}</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated repository connection could not be found.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify a different repository connection and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REP01047</th>
<th>Cannot find data source: {0}</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated data source could not be found.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Specify a different data source and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REP01049</th>
<th>Object name cannot exceed {0} characters.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated object name exceeds the maximum character limit.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Shorten the indicated object name and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REP01050</th>
<th>Object path cannot exceed {0} characters.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated object path exceeds the character limit.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Shorten the indicated object path and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REP01051</th>
<th>Cannot connect to repository &quot;{0}&quot; because it is used by another instance of DataQuant. Select different repository.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>You can not connect to the indicated repository because it is being used by another instance of the application.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Select a different repository connection and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REP01052</th>
<th>Data Source plug-in &quot;{0}&quot; was not properly initialized.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated plug-in was not properly initialized.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Change the plug-in parameters.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RGEN1000</th>
<th>An error has occurred while generating the report: {0}</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated error occurred while the report was being generated.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Resolve the indicated error and retry to generate the report.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RGEN1001</th>
<th>The action was cancelled.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The specified action was cancelled</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Rerun the specified action.</td>
</tr>
</tbody>
</table>
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 cannot 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 cannot 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 DataQuant.

RPT00044  Unable to read visual dashboard format file.
Explanation: [{CONFIG_DIR}/converter/repository/world.xml can not be found or read.
User response: Reinstall DataQuant.

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 DataQuant.

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.
Configuration directory has not been specified.
Explanation: The configuration directory for the converter can not be found.
User response: Reinstall DataQuant.

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.

Configuration directory has not been found.
Explanation: The configuration directory for the converter can not be found.
User response: Reinstall DataQuant.

Configuration file {0} has not been found.
Explanation: The configuration file for the converter can not be found.
User response: Reinstall DataQuant.

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.

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.

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.

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.

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.

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.

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.

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.

Cannot parse configuration file {0}.
Explanation: An error occurred while attempting to parse the converter configuration files.
User response: Reinstall DataQuant.

Visual Dashboard for a connection is not specified.
Explanation: The Visionary World file was not loaded properly.
User response: Reload the Visionary World file.

Invalid color description has been found.
Explanation: Corrupted tags were encountered while loading a ColorMap or ColorSequence.
User response: Reload the Visionary World file.

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 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.

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Explanation</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPT20001</td>
<td>Unable to access repository during Visual Report conversion.</td>
<td>The repository could not be accessed while the Visual Report was being converted.</td>
<td>Collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
<tr>
<td>RPT20002</td>
<td>Source visual report file has not been specified.</td>
<td>The source visual report file has not been specified.</td>
<td>Specify the source visual report file.</td>
</tr>
<tr>
<td>RPT20003</td>
<td>Source visual report file has not been found.</td>
<td>The source visual report file could not be found.</td>
<td>Specify a valid source visual report file.</td>
</tr>
<tr>
<td>RPT20004</td>
<td>Visual Report name has not been specified.</td>
<td>The visual report name has not been specified.</td>
<td>Specify a name for the visual report.</td>
</tr>
<tr>
<td>RPT20005</td>
<td>Unable to create visual report.</td>
<td>An error occurred while creating the visual report.</td>
<td>Reload the visual report.</td>
</tr>
<tr>
<td>RPT20006</td>
<td>Unable to open an embedded image.</td>
<td>An embedded image file was not found.</td>
<td>Reload the visual report.</td>
</tr>
<tr>
<td>RPT20007</td>
<td>Unknown section type: [0]</td>
<td>The structure of a section set and type was not found while loading the visual report.</td>
<td>Reload the visual report.</td>
</tr>
<tr>
<td>RPT20008</td>
<td>Unable to create Visual Report page component.</td>
<td>An error occurred while parsing and creating a page component.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Explanation</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPT20009</td>
<td>Internal error: unable to access Visual Report.</td>
<td>An internal error occurred.</td>
<td>Reload the visual report.</td>
</tr>
<tr>
<td>RPT20011</td>
<td>Unable to set a line style for a line control.</td>
<td>An undefined line style was encountered while loading the visual report.</td>
<td>Reload the visual report.</td>
</tr>
<tr>
<td>RPT20012</td>
<td>Invalid linked picture alias has been found in Visual Report.</td>
<td>A link to a picture without a name was encountered while parsing the visual report data.</td>
<td>Reload the visual report.</td>
</tr>
<tr>
<td>RPT20013</td>
<td>Unable to create Visual Report control.</td>
<td>An error occurred while loading a primitive or control object.</td>
<td>Reload the visual report.</td>
</tr>
<tr>
<td>RPT20015</td>
<td>Invalid parameter name &quot;[0]&quot; has been found.</td>
<td>The parameter value is invalid.</td>
<td>Specify a different name for the parameter indicated.</td>
</tr>
<tr>
<td>RPT20016</td>
<td>Invalid parameter value &quot;[0]&quot; has been found.</td>
<td>The parameter value is invalid.</td>
<td>Specify a different value for the parameter indicated.</td>
</tr>
<tr>
<td>RPT30001</td>
<td>Navigate to Embedded Scene: Embedded Scene not found for name &quot;[0]&quot;.</td>
<td>The Navigate to Embedded Scene action failed because the indicated embedded scene could not be found.</td>
<td>Correct the name of the embedded scene that is associated with the Navigate to Embedded Scene action, or specify another.</td>
</tr>
</tbody>
</table>
### RPT30002

**Explanation:** The Navigate Back action failed, possibly because the navigation history is empty.

**User response:** Reload the visual project and try again.

### RPT30021

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help > Collect Support Data and forward it to customer support.

### RPT30022

**Explanation:** An unknown component was encountered during runtime rendering.

**User response:** Reload the visual project.

### RPT30024

**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

**Explanation:** An error occurred while running the substitution rule to change a template.

**User response:** Check the Height property for errors.

### RPT30026

**Explanation:** An error occurred while running the substitution rule to change a template.

**User response:** Check the Width property for errors.

### RPT30028

**Explanation:** The bundle for the resource cannot be found.

**User response:** Restart the application. If this does not fix the problem, reinstall the application.

### RPT30029

**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

**Explanation:** An internal error occurred.

**User response:** Collect support data by selecting Help > Collect Support Data and forward it to customer support.

### RPT30038

**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

**Explanation:** An error occurred while creating a ClusterGraph, OrganizationChart, or TreeChart.

**User response:** Reconnect to the repository and reload the visual project.

### RPT30042

**Explanation:** The value of an axis’sPercentageLength property is invalid.

**User response:** Check the PercentageLength property value of the axis for errors.

### RPT30044

**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

**Explanation:** An invalid style property was specified for a ComboBox object.

**User response:** Check the style property for the ComboBox object for errors.

### RPT30047

**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, HOLLOWtriangle, HOLLOWTRIANGLE, HOLLOWTRIANGLE2, HOLLOWSTAR, HOLLOWPLUS, SOLIDCIRCLE, SOLIDDIAMOND, SOLID SQUARE, SOLID DIAMOND, SOLID TRIANGLE, SOLID TRIANGLE2, 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, HOLLOWtriangle, HOLLOWTRIANGLE, HOLLOWTRIANGLE2, HOLLOWSTAR, HOLLOWPLUS, SOLIDCIRCLE, SOLIDDIAMOND, SOLID SQUARE, SOLID DIAMOND, SOLID TRIANGLE, SOLID TRIANGLE2, 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.

RPT30067 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.

RPT30068 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.

RPT30069 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.

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 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.

RPT30072 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.

RPT30073 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.

RPT30074 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.

RPT30075 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.

RPT30076 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.

RPT30077 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.
2. Query "/[0]\ does not contain column named "/[1]\ - The indicated column does not exist in the indicated query.

**User response:**
1. Check the global parameter name for errors or specify a different global parameter.
2. 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.
<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Explanation</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSE01000</td>
<td>{0}</td>
<td>The indicated error occurred while accessing the repository security provider.</td>
<td>Resolve the indicated error and try again.</td>
</tr>
<tr>
<td>RSE04000</td>
<td>System internal error occurred, please check the error log.</td>
<td>A system internal error occurred.</td>
<td>Check the error log by selecting Output from the context menu of the Window menu.</td>
</tr>
<tr>
<td>RSE04001</td>
<td>Cannot find User {0}</td>
<td>The indicated user can not be found.</td>
<td>Check the name of the user for errors and try again.</td>
</tr>
<tr>
<td>RSE04002</td>
<td>User {0} already exists;</td>
<td>The indicated user already exists.</td>
<td>Specify a different user.</td>
</tr>
<tr>
<td>RSE04003</td>
<td>Group {0} already exists;</td>
<td>The indicated group already exists.</td>
<td>Specify a different group.</td>
</tr>
<tr>
<td>RSE04004</td>
<td>Cannot find group {0}</td>
<td>The indicated group can not be found.</td>
<td>Specify a different group.</td>
</tr>
<tr>
<td>RSE04005</td>
<td>Not logged in with security provider.</td>
<td>You attempted to access the repository without logging in.</td>
<td>Login to the repository and try again.</td>
</tr>
<tr>
<td>RSE04006</td>
<td>The user ID and password are invalid</td>
<td>The user ID and password that were entered are invalid.</td>
<td>Check the user ID and password for errors and try again.</td>
</tr>
<tr>
<td>RSE04007</td>
<td>Name &quot;{0}&quot; is in conflict with the existing name.</td>
<td>The new name that was entered already exists.</td>
<td>Select a different name.</td>
</tr>
<tr>
<td>RSE04008</td>
<td>Object access denied</td>
<td>Access to the specified object is denied.</td>
<td>Select a different object to access or contact your administrator for more information.</td>
</tr>
<tr>
<td>RSE04011</td>
<td>Cannot remove User {0} from Group {1}</td>
<td>The indicated user can not be removed from the indicated group.</td>
<td>Select another user to be removed from the group.</td>
</tr>
<tr>
<td>RSE04012</td>
<td>Cannot delete User {0}</td>
<td>The indicated user can not be deleted.</td>
<td>Select another user to be deleted.</td>
</tr>
<tr>
<td>RSE04013</td>
<td>Cannot delete Group {0}</td>
<td>The indicated group can not be deleted.</td>
<td>Select another group to be deleted.</td>
</tr>
<tr>
<td>RSE04014</td>
<td>Cannot initialize security provider,{0}</td>
<td>An error occurred while initializing the repository.</td>
<td>Change the repository initialization parameters and try again.</td>
</tr>
<tr>
<td>RSE04015</td>
<td>This security provider is a read-only one.</td>
<td>You can not change the principals of a read-only security provider.</td>
<td>Choose another security provider to modify.</td>
</tr>
<tr>
<td>RSE04016</td>
<td>Cannot find any name associates with UID {0} in Repository</td>
<td>The indicated repository identifier is invalid.</td>
<td>Specify a valid repository identifier and try again.</td>
</tr>
<tr>
<td>RSE04017</td>
<td>AuthType is not anonymous(&quot;none&quot;),please specify principal and credentials</td>
<td>An error occurred while initializing the LDAP security provider.</td>
<td>Change the repository initialization parameters and try again.</td>
</tr>
</tbody>
</table>

Appendix C. Messages 413
RSE04018  LDAP Configuration contains illegal values or some required values are missing, please check the preferences.

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: You 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 service 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.
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Explanation</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSE04033</td>
<td>You can not change the default Administrator's membership.</td>
<td>The membership of the default administrator can not be changed.</td>
<td>Specify the membership of a different administrator to change.</td>
</tr>
<tr>
<td>RSE04035</td>
<td>You can not rename the Administrators group.</td>
<td>The group of the default administrator can not be renamed.</td>
<td>Specify the group of a different administrator to be renamed.</td>
</tr>
<tr>
<td>RSE04036</td>
<td>Password has expired and is no longer valid.</td>
<td>The specified password has expired and is no longer valid.</td>
<td>Specify an updated password and try again.</td>
</tr>
<tr>
<td>RSE04037</td>
<td>Group could not became its own parent.</td>
<td>A group may not be its own parent. You must select a different group as the parent.</td>
<td>Select a different group to be the parent of the specified group.</td>
</tr>
<tr>
<td>SPL01001</td>
<td>Cannot find objectClass attribute of (0).</td>
<td>The indicated object does not have an objectClass attribute. Every object in the LDAP directory must contain this attribute.</td>
<td>Specify an objectClass attribute for the indicated object.</td>
</tr>
<tr>
<td>SPL01002</td>
<td>The returned entry (0) is not acceptable for LDAP Security Provider, its objectClass is not (1) or (2).</td>
<td>The indicated object is defined as neither a user or group in the LDAP security provider.</td>
<td>Edit the objectClass attribute of the indicated object to specify whether it is a user or group.</td>
</tr>
<tr>
<td>SPL01003</td>
<td>Cannot read the Group's name (attribute: (0)).</td>
<td>The LDAP directory does not contain an object that has the required attribute.</td>
<td>Edit the groupName attribute so that it will point to an existing object in the LDAP directory.</td>
</tr>
<tr>
<td>SPL01004</td>
<td>Cannot read the User's name (attribute: (0)).</td>
<td>The LDAP directory does not contain an object that has the required attribute.</td>
<td>Edit the userName attribute so that it will point to an existing object in the LDAP directory.</td>
</tr>
<tr>
<td>SPL01005</td>
<td>LDAP server host or port was not specified.</td>
<td>The LDAP server host or port was not specified.</td>
<td>Specify the LDAP server and host.</td>
</tr>
<tr>
<td>SPL01006</td>
<td>The base distinguish name of the LDAP server must be present.</td>
<td>The base distinguished name of the LDAP server was not specified.</td>
<td>Specify the base distinguished name of the LDAP server.</td>
</tr>
<tr>
<td>SPL01007</td>
<td>The distinguish name of user principal must be present.</td>
<td>The distinguished name of the principal user was not specified.</td>
<td>Specify the distinguished name of the principal user or use anonymous authentication.</td>
</tr>
<tr>
<td>SPL01008</td>
<td>The credentials of user entry must be specified.</td>
<td>The credentials of the entry user were not specified.</td>
<td>Specify the credentials of the entry user or use anonymous authentication.</td>
</tr>
<tr>
<td>SPL01009</td>
<td>Cannot create LDAP entry for administrator user: (0).</td>
<td>The user who is associated with the administrative account could not be found in the LDAP directory.</td>
<td>Specify the administrative user of the LDAP directory.</td>
</tr>
<tr>
<td>SPL01010</td>
<td>Cannot create LDAP entry for administrators group: (0).</td>
<td>The group that is associated with the administrative account could not be found in the LDAP directory.</td>
<td>Specify the administrative group of the LDAP directory.</td>
</tr>
</tbody>
</table>
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 sqjl.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]"
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated error occurred while the user rights on the indicated package were checked.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Resolve the indicated error and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02021</th>
<th>Inconsistent profile data for queries package &quot;[0]&quot; were detected. Rebind package.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>Inconsistent profile data for the indicated queries package was detected.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Rebind the package.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02100</th>
<th>The required parameter &quot;[0]&quot; is missing.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated required parameter is missing.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Collect support data by selecting Help &gt; Collect Support Data and forward it to customer support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02101</th>
<th>An error occurred while initializing the connection. [0]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated error occurred while the connection was initialized.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Resolve the indicated error and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02102</th>
<th>[0]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated database access error occurred.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Resolve the indicated error and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02103</th>
<th>An error occurred while loading package description class &quot;[0]&quot;. [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated error occurred while the package description class was loaded.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Resolve the indicated error and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02104</th>
<th>An error occurred while loading the profile for package &quot;[0]&quot;. [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated error occurred while the profile for the indicated package was loaded.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Resolve the indicated error and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02105</th>
<th>Static mode profile for package &quot;[0]&quot; is not found.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The static mode for the indicated package was not found.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Reinstall the application and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02106</th>
<th>An error occurred while creating directory &quot;[0]&quot;.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The temporary directory for the customization process could not be created.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Ensure that the application has permissions to create the directory and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02107</th>
<th>An error occurred while creating profiles directory &quot;[0]&quot;.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>An error occurred while the indicated profile's directory was created.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Ensure that the application has permissions to create the directory and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02108</th>
<th>An error occurred while saving the profile for package &quot;[0]&quot;. [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated error occurred while the profile for the indicated package was saved.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Resolve the error and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02109</th>
<th>An error occurred while saving the packages description file. [0]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated error occurred while the package description file was saved.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Resolve the error and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02110</th>
<th>An error occurred while starting customization. [0]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated error occurred while customization was started.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Resolve the error and try again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQM02111</th>
<th>An error occurred while customizing package &quot;[0]&quot;. [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong></td>
<td>The indicated error occurred while the indicated package was customized.</td>
</tr>
<tr>
<td><strong>User response:</strong></td>
<td>Resolve the indicated error and try again.</td>
</tr>
</tbody>
</table>
### 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:** Send an error report to the developers.

### SYNERRCD0001
DSS header length less than 6.

**Explanation:** An internal error in the driver code occurred.

**User response:** Send an error report to the developers.

### SYNERRCD0002
DSS header length does not match the number of bytes of data found.

**Explanation:** An internal error in the driver code occurred.

**User response:** Send an error report to the developers.

### SYNERRCD0003
DSS header C-byte not D0.

**Explanation:** An internal error in the driver code occurred.

**User response:** Send an error report to the developers.

### SYNERRCD0004
DSS header f-bytes either not recognized or not supported.

**Explanation:** An internal error in the driver code occurred.

**User response:** Send an error report to the developers.

### SYNERRCD0005
DSS continuation specified but not found.

**Explanation:** An internal error in the driver code occurred.

**User response:** Send an error report to the developers.

### SYNERRCD0006
DSS chaining specified but no DSS found.

**Explanation:** An internal error in the driver code occurred.

**User response:** Send an error report to the developers.

### SYNERRCD0007
Object length less than four.

**Explanation:** An internal error in the driver code occurred.

**User response:** Send an error report to the developers.

### SYNERRCD0008
Object length does not match the number of bytes of data found.

**Explanation:** An internal error in the driver code occurred.

**User response:** Send an error report to the developers.
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Explanation</th>
<th>User Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNERRCD0009</td>
<td>Object length greater than maximum allowed.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0010</td>
<td>Object length less than minimum required.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0011</td>
<td>Object length not allowed.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0012</td>
<td>Incorrect large object extended length field.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0013</td>
<td>Object codepoint index not supported.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0014</td>
<td>Required object not found.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0015</td>
<td>Too many command data objects sent.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0016</td>
<td>Mutually-exclusive objects present.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0017</td>
<td>Too few command data objects sent.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0018</td>
<td>Duplicate object present.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0019</td>
<td>Invalid request correlator specified.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0020</td>
<td>Required value not found.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0021</td>
<td>Reserved value not allowed.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0022</td>
<td>DSS continuation less than or equal to two.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0023</td>
<td>Objects not in required order.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
<tr>
<td>SYNERRCD0024</td>
<td>DSS chaining bit not 1, but DSSFMT bit 3 set to 1.</td>
<td>An internal error in the driver code occurred.</td>
<td>Send an error report to the developers.</td>
</tr>
</tbody>
</table>
SYNERRCD0025  Previous DSS indicated current
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.

SYNERRCD0026  Previous DSS indicated current
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.

SYNERRCD0027  DSS chaining bit not 1, but error
continuation requested.

Explanation:  An internal error in the driver code
occurred.

User response:  Send an error report to the developers.

SYNERRCD0028  Mutually-exclusive parameter
values specified.

Explanation:  An internal error in the driver code
occurred.

User response:  Send an error report to the developers.

SYNERRCD0029  Codepoint not valid command.

Explanation:  An internal error in the driver code
occurred.

User response:  Send an error report to the developers.

SYNERRCD0030  The atmind instance variable is 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.

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 US. This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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.
19-21, Nihonbashı-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions 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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as shown below.

© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

426 Getting Started with DataQuant
Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at [http://www.ibm.com/legal/copytrade.shtml](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.

**BIRT reports.** Graphical reports authored in the Business Intelligence and Report Tool (BIRT) report designer. (DataQuant for Workstation only).

**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 DataQuant 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 DataQuant 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 on 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 DataQuant interface and the database. The DataQuant 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 DataQuant 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 DataQuant 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. DataQuant uses JDBC to connect to all database repositories and data sources. DataQuant does not include JDBC drivers. The location of the JDBC drivers that DataQuant will use to connect to repositories and data sources must be defined. DataQuant 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 DataQuant files that can be imported to set up scheduled jobs.

job scheduler. DataQuant has the ability to schedule a procedure object to be executed on a periodic, recurring basis or at a specific date and time. A DataQuant procedure can perform one or more sequential steps, such as run queries, apply report templates, export reports to PDF files, or send reports using email. You can schedule your procedure to run at any given time using your local scheduler or the DataQuant 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 z/OS and 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 DataQuant for Workstation. This type of connection is not applicable to DataQuant for WebSphere users.

**OLAP queries.** OLAP queries enable users to dynamically access different dimensions of multidimensional data. Using DataQuant for Workstation and DataQuant 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 DataQuant 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. (DataQuant 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 (DataQuant for Workstation or DataQuant 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.** DataQuant 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 jobs.** You can schedule your procedure to run at any given time using your local scheduler or the DataQuant for WebSphere scheduler. Procedures that have been scheduled to run at another time are referred to as scheduled jobs.

**shared repositories.** Shared repositories are located on a database and can be shared by many users. Only DataQuant 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 DataQuant 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 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 input 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 DataQuant 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. DataQuant 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 DataQuant 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 DataQuant for Workstation, DataQuant for WebSphere, or Lotus Notes® 8.

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 DataQuant 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 DataQuant for Workstation. This type of connection does not apply to DataQuant 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 DataQuant 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 DataQuant for Workstation, you can use the Zoom option to view in hexadecimal or binary the contents of a cell in the query results.
Index

Special characters
/Batch 340
/IDisplay 340
/OObject:“ObjectOwner”.ObjectName” 340
/OPassword:password 340
/IServer:servername 340
/UserID:UserID 340
/Connection:connectionname 340
/Object:pathtoobject 340
/Password:password 340
/User:r userID 340
/&&Variable 292, 294, 307
&an 143
&CALCID 143
&GLOBAL 143
&HTML Variables 143
&LINE 143
&DATE 143
&Global Variables 143
&BOTTOMMARGIN 305
&new for break 133
&new page for footing 133
&new page for heading 133
&Repeat detail heading 133
&Blank lines after footing 133
&Blank lines after heading 133
&Blank lines before footing 133
&Blank lines before heading 133
&Break footing text 133
&Break heading text 133
&Break footing text 133
&Break heading text 133
&Blank lines before footing 133
&Blank lines after heading 133
&Repeat detail heading 133
&Break footing text 133
&Break heading text 133
&Break footing text 133
&Break heading text 133
&Break footing text 133
&Break heading text 133
&Break footing text 133
&Break heading text 133
&Break footing text 133
&Break heading text 133
&Break footing text 133
&Break heading text 133
Breaks formatting options

A
accessibility 353
  assistive technology 354
  keyboard equivalents 353
  keyboard navigation 353
  navigation in WebSphere 356
  navigation in Workstation 354
  operating system 353
  ACROSS 144
  ACTION 300, 308
  Add Calculated Column 122
  Add to Favorites 22
  Add to Startup 22
  adding an Execute JavaScript event how to 211
  Administrator perspective 2, 3
  aggregations 165
  Analytical queries 84, 87
    Analytical queries 87
    append queries 85
    conditional grouping 90
    crosstab queries 86
    join queries 86
    normalize 88
  ANCHORn 135
  Appearance preferences 11, 13
  append queries 85
  ATTACHMENT 301
  AutoFit 122
  Average 122
  AVERAGE 144
  axes chart format options 169

B
best fit models 151
binary 121
binding resource limits 348
BIRT reports 3, 127, 147
BODY 301
BOLD 305
Boolean 194
Border 201
BOTTOM 291
BOTTOMMARGIN 305
Break Footing 143
Break Heading 143
Breakn 144
BREAKnX 144
breakpoints 272
Breaks formatting options
  Blank lines after footing 133
  Blank lines after heading 133
  Blank lines before footing 133
  Blank lines before heading 133
  Break footing text 133
  Break heading text 133
  New page for break 133
  New page for footing 133
  Put break summary at line 133
  Repeat detail heading 133

C
cache resource limits 351
caching 46
  repository data 46
  CALCid 144
calculated columns 106
  calculated expressions 230
  Calculation Expression 143
  Calculations formatting options
    Edit 136
    Expression 136
    ID 136
    Pass nulls? 136
    Width 136
  Category Axis 165
  category chart format options 168
  CCLIST 301
  CCSID 121, 296
  Change favorite name 28
  Change Startup Name 28
  character data edit codes 327
  CHARSET 305
  chart area chart format options 169
  charts 163
    Display editor 164
    result set grid 164
  Cheat Sheets view 3
  classic reports 127
  client area chart format options 173
  Color 194
  color maps 271, 272
color maps (continued)
  creating 272
  using 273
color sequences 271, 274
  creating 274
  using 275
Column Expression 143
Columns formatting options
  Edit 136
  Heading 136
  Indent 136
  Num 136
  Seq 136
  Usage 136
  Width 136
Columns with Aggregates 165
command bar 341
command line 339
  parameters 340
CommandLine 295
COMMENT 300, 308
comments, sending to IBM 340
compile visual dashboard 222
compound condition filter 104
Condition 143
conditional grouping 90
Conditions formatting options
  Expression 138
  ID 138
  Pass nulls 138
CONFIRM 295, 296, 300, 307, 308
CONNECT
  PASSWORD 291
  ServerName 291
  UserName 291
Connectivity 232
  connectivity tool 235
connector objects
  Connection Point 248
  Elbow Connector 248
  Spline Connector 248
  Straight Connector 248
Connectors 232
container objects
  EmbeddedContent 250
  EmbeddedScene 250
Containers 232
data assist 55
control objects
  Button 243
  CheckBox 243
  Combo 243
  DateTime 243
  HorzSlider 243
  ListBox 243
  RadioGroup 243
  Slicer 243
  TextBox 243
  VertSlider 243
Controls 232
customizations for highlighting xi
CONVERT
  QueryName 292
Convert to HTML 149
Convert to Visual Report 149
COPIES 305
Copy 22
correct format 52
Count 122
COUNT 144
CPCT 144
Create Cyclicality 151
Create New Visual Dashboard wizard tabs
  Local 188
  Predefined 188
  Repository 188
Create New Visual Report wizard tabs
  Local tab 177
  Predefined tab 177
  Repository tab 177
Create Performance 151
Create Seasonality 151
Create Trend 151
crosstab queries 86
CSUM 144
cube data 79
Cube Structure tree 79, 81
cumulative forecast error 159
Cumulative Percentage of Group 122
Cumulative Percentage of Total 122
Cumulative Sum 122
curve fitting 158
Curve fitting methods 151
Custom 232
customizing perspectives 10

D

Data Explorer view 3
Data Management preferences 11, 14
data source connections 191
Data Source Connections view 3
Data Source Object 149
data source options
  embedded 153
  linked 153
data sources 146
  listing objects 129
data symbol objects
  Candlestick 246
  EventBand 246
  HorizontalLabelBar 246
  HorizontalValueBar 246
  Marker 246
  PriceIntervalBar 246
  ThresholdBand 246
  VerticalLabelBar 246
  VerticalValueBar 246
Data Symbols 232
data templates 269
  visibility 270
data types 230
Database folder 22
DATAFORMAT 296
DataQuant Calculator view 3
date data edit codes 328
date parameters
  date column 153
  from 153
  number of future periods 153
  period 153
DateTime 194
DA 
TETIME 305
DateTimeSet 194
DB2 for z/OS 1
DB2 passwords 46
Default editor 28
Delete 22
Design mode 225
detail 97
Detail Block 143
Detail Heading 143
Details formatting options
  Blank lines after block 138
  Detail block text 138
  Detail heading text 138
  Enable 138
  Include column headings with detail headings 138
  Keep block on page 138
  New page for detail block 138
  Put tabular data at line 138
  Repeat detail heading 138
Diagram Query editor 64, 66
  adding tables 58, 69
  join conditions 69
directly importing an IXF file into a database table 118
DISPLAY
  &&Variable 292
  ObjectName 292
Display editor 163
Display editor panes
  Display modes 163
  Layout structure 163
  Result set 163
Display Excel Sheet 122
  display modes 163, 166
  adding 166
  editing chart format 167
    axes 169
    category 168
    chart area 169
    client area 173
    gridlines 171
    labels 171
    legend 173
    plot area 173
    range 172
    series 168
    title 172
    value(Y)series 169
    X-Axis 170
    Y-Axis 170
  editing chart types 167
  editing data series 167
  Display modes pane 163, 166
  Display Report 122
  Display Report wizard 127
display settings
  Columns 234
  Details 234
  Icons Only 234
  List 234
double exponential smoothing 157
Double Exponential Smoothing 151
DRAW
  TableName 294
  TYPE 294
  Draw Query editor 70
    listing data source tables 71
  Drill Down 122
  Drill Up 122
  drilldown 195
  Drilldown wizard 195
driver class name 15, 36
DSQAO 321
DSQCP 326
DSQDC 325
DSQEC 323
DSQQW 316
Dynamart Filter Wizard 104
dynamic SQL 1
E
Eclipse 1
EDIT
  &&Variable 294
  ObjectName 294
edit codes 327
  character data 327
  custom codes for classic reports 333
  date data 328
  graphic data 329
  metadata 338
  numeric data 329
  samples 334
  time data 331
  timestamp data 332
  user defined 333
editing database tables 32
editor window 225
e-mail
  query results 110
  embedded scenes 201
  parameterized 202
End Session 217
ERASE
  CONFIrm 295
  ObjectName 295
event actions 203
  assigning 204
  End Session 217
  Execute JavaScript 211
  Execute procedure 204, 210
  Execute shell command 204, 214
  Execute SQL statement 204, 209
  Export to excel 215
  Export to Excel 204
  Export to PDF 218
  Invalidate Query Cache 220
  Jump to new location 204, 205
  Launch LOB 220
  Navigate back 204, 209
  Navigate to an embedded scene 208
  Navigate to embedded scene 204
  Open URL 221
  Print scene 204, 211
  Refresh Component 219
  Reload scene 217
  Send Email 212
  Send Event to Embedded Scene 213
  Set focus 216
  Set values 204, 206
  Show message 204, 214
EventInterval 236
Form structure tree (continued)
Conditions 130
Details 130
Final 130
Main 130
Options 130
Page 130
form variables
&an 143
&CALCid 143
&COUNT 143
&DATE 143
&Global Variables 143
&HTML Variables 143
&n 143
&PAGE 143
&ROW 143
&TIME 143
Format 122
formatting options
breaks
Blank lines after footing 133
Blank lines after heading 133
Blank lines before footing 133
Blank lines before heading 133
Break footing text 133
Break heading text 133
New page for break 133
New page for footing 133
Put break summary at line 133
Repeat detail heading 133
calculations
Edit 136
Expression 136
ID 136
Pass nulls? 136
Width 136
columns
Edit 136
Heading 136
Indent 136
Num 136
Seq 136
Usage 136
Width 136
conditions
Expression 138
ID 138
Pass nulls 138
details
Blank lines after block 138
Detail block text 138
Detail heading text 138
Enable 138
Include column headings with detail headings 138
Keep block on page 138
New page for detail block 138
Put tabular data at line 138
Repeat detail heading 138
final
Blank lines before text 141
New page for final text 141
Put final summary at line 141
Text 141
main
Edit 131
Heading 131

formatting options (continued)
main (continued)
Indent 131
Seq 131
Total width of report columns 131
Usage 131
Width 131
options
Across heading separator 140
Across summary column 140
Automatic reordering of report columns 140
Break summary separator 140
Column heading separator 140
Column wrapped lines kept on a page 140
Default break text (*) 140
Detail spacing 140
Final summary separator 140
Function name in column heading when grouping 140
Line wrapping width 140
Number of fixed columns in report 140
Outlining for break columns 140
Page renumbering at the highest break level 140
Place LOB contents inline 140
Report text line width 140
page
Blank lines after footing 142
Blank lines after heading 142
Blank lines before footing 142
Blank lines before heading 142
Page footing text 142
Page heading text 142
forms 130
designing 130
saving with data source 147
saving without data source 147
FORWARD 300
free style condition filter 106
From 301
From data source 128
From file 128
From open object 128
From repository 128
full outer join 59, 69

G
General preferences 11, 12
getting started 1
global average 161
global parameters 276
adding 278
ExternalEventID 277
SendExternalEvent 277
Translation Table 277
using 279
ViewerHeight 277
ViewerWidth 277
ViewerX 277
ViewerY 277
ViewerZoom 277
Global parameters 227
global resources
color maps 271, 272
creating 272
using 273
color sequences 271, 274
creating 274
global resources (continued)
color sequences (continued)
  using 275
  expiration schedules 271
global parameters 276, 277
  adding 278
  using 279
images 271
  parameters 271
security lists 271, 279
stock images 275
  creating 276
  using 276
translation tables 271, 280
  switching 281
global trend 160
global variables 315, 316
  DSQAO 321
  DSQCP 326
  DSQDC 325
  DSQEC 323
  DSQQW 316
  system 316
user 315
Global Variables preferences 11
Go to 122
Grab 232
grand total 97
graphic data edit codes 329
gridlines chart format options 171
GROUP 144
grouping and aggregation 107, 165
  fields 108
Grouping and Aggregation
  Average 122
  Count 122
  Cumulative Percentage of Group 122
  Cumulative Percentage of Total 122
  Cumulative Sum 122
  First 122
  Last 122
  Maximum 122
  Minimum 122
  No Aggregation 122
  Percentage of Group 122
  Percentage of Total 122
  Side Group 122
  Standard Deviation 122
  Sum 122
  Top Group 122
  grouping hierarchy options
    available columns 154
    selected columns 154

H
header 97
help content 15
Help preferences 11, 14
Help view 3
hexadecimal 121
Hierarchical filter 82
highlighting conventions xi
Holt-Winters method 151, 157
host 15
HP-UX 1
HTML 225

HTML variable 135

I
ICatalogPassword:catalogpassword 340
ICatalogUserID:cataloguserID 340
icons
  Create Cyclicality 151
  Create Performance 151
  Create Seasonality 151
  Create Trend 151
  Run Query 151
  Show Forecast Outline Tree 151
IDENTIFIER 294
IMAGEn 135
images 271
import
  CSV 115
    result set structure 116
    separator options 116
  legacy Visionary world 190
  legacy visual reports 179
  table data 118
  tables 118
  TXT 115
    result set structure 116
    separator options 116
IMPORT
  ACTION 300
  COMMENT 300
  CONFIRM 300
  FileName 300
  LANGUAGE 300
  LOBSFROM 300
  ObjectName 300
  SHARE 300
import CSV 115
  result set structure 116
  separator options 116
import TXT 115
  result set structure 116
  separator options 116
importing data contained in ixf files 117
information zooming 196
  layout objects 197
  scenes 198
Informix 1
inner join 59, 69
Insert Query window 192
Integer 194
Invalidate Query Cache 220
ITALIC 305

J
JARS 15, 36
JDBC 1
JDBC drivers 1, 35
JDBC libraries 15, 36
JDBC Libraries preferences 11
JDBC library settings file 35
JDBC preferences 15, 36
JDF 287
job definition files 287
job files 287
join queries 86
labels chart format options 171
LANGUAGE 296, 300, 308
Last 122
LAST 144
Launch LOB 220
Layers view 3, 175, 271
layout objects
  Bar chart 251
  BarChart 255
    extended 256
  OLAP 256
  Candlestick chart 251
  CandlestickChart 261
  Cluster graph 251
  ClusterGraph 262
  Column chart 251
  ColumnChart 255
    extended 256
  OLAP 256
  data symbols 266
    graphic objects 267
    data templates 268
      new 268
default connectors 266
  Dial 265
  Event Band chart 251
  EventBandChart 262
  extending functionality 266
  Grid 251
Horizon 251
information zooming 197
Linear map 251
LinearMaps 263
Matrix 251, 263
multiple query results 267
Multivariate chart 251
MultivariateChart 259
nesting 270
  Organization chart 251
  OrganizationChart 264
Pie chart 251
  PieChart 257
    extended 258
      OLAP 258
queries 265
query parameters 266
  Scatter chart 251
  ScatterChart 260
Simple form 251
Spiral 251
Stock chart 251
StockChart 260
Table 251, 252
  populated 253
    simple 252
  summary row formatting 254
    summary rows 254
Timeline chart 251
TimelineChart 262
Tree chart 251
TreeChart 264
XY chart 251
XYChart 260

Layout page 79
layout properties
  Conditional 97
  Font 97
  Format 97
  General 97
Layout Properties dialog 97
  conditional formatting options 103
    font fields 100
    font formatting options 100
    format fields 102
    formatting options 101
  general column formatting 99
Layout structure pane 163, 165
Layout Structure tree 107
Layout tab 81
Layout view 81
Layouts 232
left outer join 59, 69
LEFTMARGIN 305
legacy Visionary worlds importing 190
legacy visual reports 178
  importing 179
    opening 179
legend chart format options 173
LENGTH 296, 305
Level of Detail 197, 198
LIMIT LOCAL
  Value 301
  VariableName 301
limits resource limits 344
linear trend 160
LINKn 135
links
  non-IBM Web sites 427
Linux 1
Literal 194
literals 230
LOB data 119
LOB options resource limit
  overriding 121
LOB options resource limits 350
LOBFILE 296
LOBs preferences 11, 17
LOBSFROM 300
LOBSINFILE 296
LOBSTO 296
local jobs 285
Local parameters 227
local trend 160
Log preferences 11, 18
LookAI xi
Lotus Notes 175

MAIL TO
  ATTACHMENT 301
  BODY 301
  CCLIST 301
  From 301
  METHOD 301
  ObjectName 301
  SMTP SERVER 301
  SUBJECT 301
  TYPE 301
MAILTON 135
Main formatting options
  Edit 131
  Heading 131
  Indent 131
  Seq 131
  Total width of report columns 131
  Usage 131
  Width 131
Mapping table 119
matching patterns 52
Maximum 122
MAXIMUM 144
MDX page 79
mean absolute deviation 159
mean absolute percent error 159
mean square error 159
measures 81
menu commands
  Personal view 28
  Workspace 22
messages
  methods for accessing xiii
metadata edit codes 338
METHOD 301, 308
Microsoft Windows 1
Minimum 122
MINIMUM 144
MinimumZoom 197, 198
Miscellaneous 232
MODE 296, 308
moving average 161
Moving Average 151
multiple result sets 122
multiplicative decomposition 158
Multiplicative Decomposition method 151

N
name filter 24
Navigate back 204, 209, 236
Navigate to an embedded scene 208
Navigate to embedded scene 204
network connections 41
network repositories 38
neural network 159
Neural Networks 151
New 22
No Aggregation 122
normalize queries 88
notices
  legal 425
Number 194
NumberSet 194
numeric data edit codes 329

O
object name 24
object owner 24
object tracking resource limits 349
ObjectName 292, 294, 295, 296, 300, 301, 305, 307, 308, 340
objects
  creating 51
data source 129
file 94

options (continued)
  opening 94
  palette 233
QMF catalog 94
repository 94
saving 92
  in a repository 92
  in the QMF catalog 93
to a file 93
OLAP folder 22
OLAP queries
  building offline 81
  building online 81
  filtering cube data 82
  modifying 81
  opening 80
retrieving cube data 79
OLAP Query menu commands
  Set Data Source 83
  Set User Information 83
OLAP query results 83
OMIT 144
Open 22
Open From 22
open perspective 10
Open URL 221
opening the contents of an IXF file 117
Options formatting options
  Across heading separator 140
  Across summary column 140
  Automatic reordering of report columns 140
  Break summary separator 140
  Column heading separator 140
  Column wrapped lines kept on a page 140
  Default break text (*) 140
  Detail spacing 140
  Final summary separator 140
  Function name in column heading when grouping 140
  Line wrapping width 140
  Number of fixed columns in report 140
  Outlining for break columns 140
  Page renumbering at the highest break level 140
  Place LOB contents inline 140
  Report text line width 140
options resource limits 345
Oracle 1
ORDER 296
ORIENTATION 305
Outline view 3, 147
Output view 3, 175
OUTPUTMODE 296
overview 1

P
Page Footing 143
Page formatting options
  Blank lines after footing 142
  Blank lines after heading 142
  Blank lines before footing 142
  Blank lines before heading 142
  Page footing text 142
  Page heading text 142
Page Heading 143
PAGENO 305
palette objects
  default property value 235
palette objects (continued)  
inserting 233
Palette view 3, 175, 232
connectivity tool 235
connector objects 248
inserting 248
container objects 250
EmbeddedContent 251
EmbeddedScene 250
control objects 243
DateTime 245
inserting 244
Slicer 246
custom objects 249
data symbol objects 246
inserting 247
display settings 234
layout objects 251
primitive objects 238
AlignmentPanel 242
lines 239
Picture 241
shapes 240
text 239
Properties view 223
Timer 236
parameter hints 56
parameter types
Boolean 194
Color 194
DateTime 194
DateTimeSet 194
FilePath 194
FontName 194
Integer 194
Literal 194
Number 194
NumberSet 194
Percentage 194
Point 194
PointSet 194
PolySet 194
Text 194
TextSet 194
parameters 271
global 276
query 276
scene 276
PASSWORD 291
Paste 22
Paste Link 22
path 15
PCT 144
PDF 225
Percentage 194
Percentage of Group 122
Percentage of Total 122
Personal view 3, 28
Personal view menu commands
Change favorite name 28
Change Startup Name 28
Remove from Favorites 28
Remove from Startup 28
perspectives 9
Administrator 2, 3
closing 10
configuring 10
perspectives (continued)
QMF 3
Report Designer 3
resetting 11
User 3, 9
Visual Designer 3, 177, 223
plot area chart format options 173
Point 194
PointSet 194
polynomial regression 158
PolySet 194
port 15
preferences
Appearance 11, 13
Data Management 11, 14
General 11, 12
Global Variables 11
Help 11, 14
JDBC 15, 36
JDBC Libraries 11
LOBs 11, 17
Log 11, 18
Report Design 11, 19
Server-side File System 18
Themes 19
Third Party Libraries 20
user 11
Visual Designer 11, 20
primitive objects
Alignment panel 238
Arrow 238
Double Arrow 238
Ellipse 238
Label 238
Line 238
MultiPolygon 238
MultiPolyline 238
Picture 238
Polygon 238
Polyline 238
Rectangle 238
Round Rectangle 238
Primitives 232
PRINT
BOLD 305
BOTTOMMARGIN 305
CHARSET 305
COPIES 305
DATETIME 305
FORM 305
ITALIC 305
LEFTMARGIN 305
LENGTH 305
ObjectName 305
ORIENTATION 305
PAGENO 305
PRINTER 305
RIGHTMARGIN 305
SIZE 305
TOPMARGIN 305
typeface 305
Point 194
USEFORMPS 305
WIDTH 305
print options 342
Print Preview mode 225
Print scene 204, 211, 236
PRINTER 305

Index 443
procedures 283
creating 283
scheduling 284
Progress view 3
Project Explorer 224
Project Explorer folders
Connections 175, 186
Globals 175, 186
Main Page 175
Queries 175, 186
Scenes 186
Project Explorer view 3, 175
prompt
shared prompts 76
simple prompts 71
SQL prompts 71
prompt hierarchies 71
creating 73, 74
editing 75
working with 74
Prompted Query editor
adding tables 58, 69
columns 56
join conditions 56
joining columns 59
joining tables 59
query results 60
row conditions 56, 62
sort conditions 56, 61
tables 56
prompts
working with 72
properties
data types 230
expression designer 228
expressions 229
elements 229
examples 231
properties (continued)
extpressions (continued)
syntax rules 230
groups 227
literal values 230
Properties 22
Properties view 3, 97, 175, 227
Property Editor view 3

Q
QMF Catalog folder 22
QMF catalog objects 94
QMF perspective 3
queries
creating 53
OLAP
building offline 81
building online 81
filtering cube data 82
modifying 81
opening 80
retrieving cube data 79
relational 53
specifying in visual applications 222
specifying in visual dashboards 222
specifying in visual reports 222
visual 54
Query Editor 28
query environment 339
Query menu commands
Add New Statement 78
Cancel 78
Comment 78
Confirm Stored Procedure Parameters 78
Convert Query to Visual Query 78
Prepare 78
Reformat Text 78
Remove Current Statement 78
Run 78
Set Data Source 78
Set Font 78
Set Row Limit 78
Set User Information 78
Transfer to 78
Uncomment 78
query parameters 276
query results 97
query editing 228
query results
query results 97
compound condition 104
free style condition 106
formatting 97
grouping and aggregation 107
LOB data 119
OLAP 83
sorting 124
Query Structure tree 79, 81
QueryName 292
Range chart format options 172
Recently Used folder 28
REFn 135
Refresh 22
Refresh Component 219
regular filter 82
regular save 125
relational queries 53
Reload scene 217
remote jobs 285, 286
remote server 15
Remove from Favorites 28
Remove from Startup 28
Rename 22
Report Design preferences 11, 19
Report Designer perspective 3
Report Designer view 147
reports 127
BIRT 147
classic data source 128
designing forms 130
forms 130
saving 146
visual 175
repositories connections 34, 37
import 37
creating connections 38
file-based 41
network 41
shared 39
web service 42
creating new objects 51
creating web links
data objects 44
editing connections 44
managing connections 44
personal 33
personal repositories 33
removing connections 44
repository connection settings 1
repository connections 34, 37
Repository Connections view 3
repository objects 94
folders 51
forms 51
links 51
OLAP queries 51
personal repositories 51
procedures 51
queries 51
queries using the Draw Query wizard 51
repository connections 51
visual dashboard template categories 51
visual dashboards 51
visual report template categories 51
visual reports 51
workspaces 51
repository objects caching 46
Reset All 122
Reset Formatting 122
RESET GLOBAL
VariableName 307
resetting perspectives 11
Resource Explorer view 3

resource limits
binding 348
cache 351
limits 344
LOB 121
LOB options 350
object tracking 349
options 345
save data 347
SQL verbs 344
timeouts 343
viewing 342
Result set pane 163, 165
Results menu commands
Add Calculated Column 122
AutoFit 122
Display Excel Sheet 122
Display Report 122
Drill Down 122
Drill Up 122
Export 122
Find 122
Font 122
Format 122
Go to 122
Grouping and Aggregation 122
Reset All 122
Reset Formatting 122
Retrieve All 122
Retrieve All LOB Values 122
Show Summaries 122
Show Total Summaries 122
Sort 122
results page 79
Results tab 81, 82
RESULTSET 312
Retrieve All 122
Retrieve All LOB Values 122
REXX Console view 283
Rich Client 225
right outer join 59, 69
RIGHTMARGIN 305
ROWIDADD 308
ROWIDDISP 308
ROWIDDISPnnn 308
ROWIDNAME 308
ROWLIMIT 307
Run 22
RUN
&$Variable 307
CONFIRM 307
FORM 307
ObjectName 307
ROWLIMIT 307
Run Query 151
Runtime mode 225

save 92
SAVE
ACTION 308
COMMENT 308
CONFIRM 308
LANGUAGE 308
METHOD 308
MODE 308
SAVE (continued)

ObjectName 308
ROWIDADD 308
ROWIDDISP 308
ROWIDDSPPnn 308
ROWIDNAME 308
SCOPE 308
SHARE 308
SPACE 308

save all 92
save at 92
save data resource limits 347
save methods
fast 125
regular 125
scene parameters 276
SceneCenter 201
scenes 193, 198
schedule types
always expires 183
at specific date 183
daily 183
monthly 183
never expires 183
weekly 183
scheduled jobs 284
restoring after logon information change 286
viewing 285
scheduling jobs 284
SCOPE 308
Search view 3
Second Sort Rule 124
security lists 271, 279
Select 232
Send Email 212
Send Event to Embedded Scene 213
SEPARATOR 296
series chart format options 168
Server-side File System
preferences 18
ServerName 291
service information xi
Set Data Source 149
Set focus 216
Set Font 149
SET GLOBAL
Value 310
VariableName 310
SET INVISIBLE
Value 310
VariableName 310
SET LOCAL
Value 311
VariableName 311
SET LOCAL WITH VALUES
Value 311
VariableName 311
Set Name Filter 22
SET OPTIONS
STOPONERROR 311
SUPPRESSMESSAGES 311
Set User Information 22, 149
Set values 204, 206, 236
SHARE 300, 308
shared repositories 38, 39, 46
SHOW
RESULTSET 312

SHOW (continued)

VIEW 312
Show Forecast Outline Tree 151
Show in Repository 22
Show message 204, 214, 236
Show Summaries 122
Show Total Summaries 122
show view 10
Side Group 122
Side groups 165
Side Groups 165
silent installation 358
simple moving average 156
simple prompts
adding 72
single exponential smoothing 156
Single Exponential Smoothing 151
SIZE 305
SMTPPASSWORD 301
SMTPSERVER 301
SMTPUSER 301
Solaris 1
Sort 122
SPACE 308
special registers 32
SPLIT 296
SQL
fonts 78
SQL page 79
SQL prompts 71
SQL Query editor 53
content assist 55
parameter hints 56
SQL Server 1
SQL verbs resource limits 344
Standard Deviation 122
Startup folder 28
static SQL 1
static SQL packages 39
STDEV 144
stock images 275
creating 276
using 276
stored procedures 39
strategy options
aggregation strategy 154
construction strategy 154
distribution strategy 154
forecast column 154
SUBJECT 301
substitution variables 77
Sum 122
SUM 144
support information xi
syntax diagrams 288
syntax diagrams, how to read xi

T

tab group 10
Table Editor 28
Prompted tab 30, 31
row conditions 31
sort conditions 30
Table Viewer 28
TCPCT 144
Text 194
Themes preferences 19
Third Party Libraries preferences 20
Third Sort Rule 124
Third-Party Libraries preferences 11
time data edit codes 331
time-series data 151
timeouts resource limits 343
Timer object 236
time-series data edit codes 332
title chart format options 172
Top Group 122
Top groups 165
Top Groups 165
TOPMARGIN 305
TPCT 144
tracking signal 159
translation tables 271, 280
switching 281
TYPE 301
Type of sort
Ascending 124
Descending 124
TYPEFACE 305

UNICODE 296
UNITS 296
URL template 15, 36
usage codes
ACROSS 144
AVERAGE 144
Breakn 144
BREAKnX 144
CALCid 144
COUNT 144
CPCT 144
CSUM 144
FIRST 144
GROUP 144
LAST 144
MAXIMUM 144
MINIMUM 144
OMIT 144
PCT 144
STDEV 144
SUM 144
TCPCT 144
TPCT 144
USE REPOSITORY
RepositoryName 313
USEFORMPS 305
user defined edit codes 333
user information 34
user interface 35
user menus
Form 149
OLAP Query menu 83
Query 78
results 122
User perspective 1, 3, 9
user preferences 11
UserName 291

VALUE 301, 310, 311
value(Y)series chart format options 169
Values 165
VariableName 301, 307, 310, 311
Variables bar 222
VIEW 312
viewpoints
modifying 200
navigation 199
views 9
Cheat Sheets 3
Data Explorer 3
Data Source Connections 3
DataQuant Calculator 3
Events 3, 175, 231
Help 3
Layers 3, 175, 271
Layout 81
opening 10
Outline 3, 147
Output 3, 175
Palette 3, 175, 223, 232
Personal 3, 28
positioning 10
Progress 3
Project Explorer 3, 175, 223, 224
Properties 3, 97, 175, 223, 227
Property Editor 3
Report Designer 147
Repository Connections 3
Resource Explorer 3
REXX Console 283
Search 3
Workspaces 3, 21
views and perspectives 9
visual applications
actions
Execute JavaScript 211
visual dashboard templates 189
visual dashboards 186
actions
End Session 217
Execute JavaScript 211
Execute procedure 210
Execute shell command 214
Execute SQL statement 209
Export to excel 215
Export to PDF 218
Invalidate Query Cache 220
Jump to new location 205
Launch LOB 220
Navigate back 209
Navigate to an embedded scene 208
Open URL 221
Print scene 211
Refresh Component 219
Reload scene 217
Send Email 212
Send Event to Embedded Scene 213
Set focus 216
Set values 206
Show message 214
adding 193
creating 188
data source connections 191
deleting 193
visual dashboards (continued)
drilldown 195
embedded scenes 201
parameterized 202
event actions 203
information zooming 196
layout objects 197
scenes 198
navigation features 195
planning 187
queries 192
renaming 193
scenes 193
parameters 194
templates 189
viewpoint navigation 199, 200
visual designer 223
editor controls 226
editor window 225
grid 226
guides 226
rulers 226
Visual Designer
preferences 20
Visual Designer perspective 3, 177
Visual Designer preferences 11
visual report main pages
Detail 181
Detail Footer 181
Detail Header 181
Page Footer 181
Page Header 181
Report Footer 181
Report Header 181
visual reports 127, 175
converting to offline 185
creating 177
data source connections 183
fixed pages 182
legacy 178
importing 179
opening 179
main pages 181
queries 180, 222
data retrieval schedules 185
templates 177
visual reports fixed pages
Fixed1 182
Fixed2 182
PageFootering 182
PageHeading 182
workspaces
adding a new workspace 27
adding content 26
creating 27
discovering contents 22
filtering data source objects 24
folders 24
linking to objects
repository 25
workspace 24
moving objects 26
properties 25
refreshing contents 26
Worksapce view 3, 21
Workspaces view menu commands
Add to Favorites 22
Add to Startup 22
Copy 22
Delete 22
Explore 22
New 22
Open From 22
Paste 22
Paste Link 22
Properties 22
Refresh 22
Rename 22
Run 22
Set Name Filter 22
Set User Information 22
Show in Repository 22
X
X-Axis chart format options 170
XLocation 199
XMLA connectivity 1
Y
Y-Axis chart format options 170
YLocation 199
Z
zoom 121
ZoomPct 201
W
web service repositories 38, 42
weighted moving average 156
Weighted Moving Average 151
WIDTH 296, 305
workspace properties
Connection Parameters 25
LDAP 25
Login mapping 25
Plug-ins 25
Resource Limits 25
Security 25

448 Getting Started with DataQuant